

Niels Seidel

# Interaction Design Patterns und CSCL-Scripts für Videolernumgebungen





Niels Seidel

## **Interaction Design Patterns und CSCL-Scripts für Videolernumgebungen**







**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

# **Interaction Design Patterns und CSCL-Scripts für Videolernumgebungen**

Dem Wissenschaftlichen Rat des Internationalen Hochschulinstitutes  
Zittau  
eingereichte

## **DISSERTATION**

zur Erlangung des akademischen Grades  
doctor rerum politicarum  
(Dr. rer. pol.)

vorgelegt von  
Niels Seidel  
geboren am 7. Oktober 1981 in Löbau

Gutachter: Prof. Dr. rer. pol. habil. Thorsten Claus (TU Dresden)  
Prof. Dr. rer. pol. habil. Bärbel Fürstenau (TU Dresden)  
Prof. Dr. rer. nat. Christian Kohls (TH Köln)

Einreichung: Zittau, den 10. November 2016

Disputation: Dresden, den 26. Februar 2018



**Niels Seidel**

**Interaction Design Patterns und CSCL-Scripts für Videolernumgebungen**

## Zusammenfassung

In den letzten Jahren haben Lernvideos im Bereich des informellen und formellen Lernens an Bedeutung gewonnen. Inwieweit Lernende bei der Nutzung von Videos unterstützt werden und Lehrende didaktische Szenarien umsetzen können, hängt jedoch von der eingesetzten Videolernumgebung ab. Es ist Anliegen der vorliegende Arbeit, Prinzipien des *User Interface Designs* sowie Komponenten und Mechanismen videobasierter Lehr-Lern-Szenarien in Bezug auf Videolernumgebungen zu identifizieren, zu beschreiben und technisch zu realisieren. Das Ziel dieser Forschungsarbeit besteht darin, Gestaltungsprinzipien in Form von Interaction Design Patterns zu erarbeiten und computergestützte videobasierte Lehr-Lern-Szenarien mit Hilfe von CSCL-Scripts durch eine geeignete Spezifikation und Formalisierung zu realisieren.

Für die Erarbeitung der Interaction Design Patterns wurden 121 Videolernumgebungen hinsichtlich 50 Kategorien in einer Inhaltsanalyse empirisch untersucht und dokumentiert. Unter Berücksichtigung ähnlicher Patterns aus thematisch verwandten Pattern Languages und den Erfahrungen aus der Implementierung und dem Einsatz von Videolernumgebungen entstanden 45 Interaction Design Patterns für verbreitete Lösungen für wiederkehrende Probleme bei der Gestaltung und Entwicklung von Videolernumgebungen. Diese Patterns wurden auf Pattern Konferenzen diskutiert und im Anschluss evaluiert sowie bei der Konzeption, Entwicklung und Bewertung mehrerer Videolernumgebungen angewendet. Zudem wurde das Software Framework *VI-TWO* vorgestellt, mit dem sich fast alle Patterns auf einfache Weise in Web-Anwendungen realisieren lassen.

Zur Spezifikation videobasierter CSCL-Scripts wurden existierende videobasierte und nicht videobasierte Scripts analysiert. Im Ergebnis unterschieden sich videobasierte CSCL-Scripts von allgemeinen CSCL-Scripts vor allem hinsichtlich der mit dem Video verknüpften oder darin verankerten Aufgaben und Aktivitäten. Videos werden dabei nicht als monolithische Einheiten, sondern als zeitkontinuierliche Medien betrachtet, in denen weitere Informationen zeitgenau verankert und Lernaktivitäten stattfinden können. Außerdem ließen sich drei Typen videobasierter CSCL-Scripts identifizieren: (1) Scripts zur Analyse und Diskussion von Videoinhalten, (2) *Peer Annotation* Scripts einschließlich dem *Peer Assessment* und (3) *Jigsaw*-Scripts, die das problembasierte Lernen mit Hilfe von Videos ermöglichen. Unabhängig davon variiert die Komplexität der Scripts auf drei Stufen: (1) sofern voneinander abgegrenzte zeitliche Phasen von Lernaktivitäten bestehen, (2) wenn darüber hinaus die Teilnehmer innerhalb von Gruppen Aufgaben bearbeiten (*intra-group*) und (3) falls außerdem Aufgaben auch gruppenübergreifend bearbeitet werden (*inter-group*). Auf Grundlage der Spezifikation konnte ein Datenmodell und ein Modell für die nutzerseitige Modellierung von Scripts verschiedener Komplexitätsstufen sowie Typen entwickelt und in dem CSCL-System *VI-LAB* realisiert werden.

Diese Arbeit leistet in zweifacher Hinsicht einen Beitrag zur Forschung im Bereich E-Learning. Zum einen beschreiben die Interaction Design Patterns wiederkehrende *User Interface* Lösungen und stellen somit ein Hilfsmittel für Designer, Software Entwickler und Lehrende bei der Gestaltung und Implementierung von Videolernumgebungen dar. Zum anderen wurden durch die Spezifikation und softwareseitige Umsetzung videobasierter CSCL-Scripts Voraussetzungen geschaffen, die den praktischen Einsatz und die weitere Untersuchung des kollaborativen Lernens mit Videos ermöglichen.

## Summary

### **Interaction design patterns and CSCL scripts for video-based learning environments.**

In the past years videos played an increasingly important role in formal and informal education. The support of video-based learning activities as well as the ability to incorporate respective instructions heavily depends on the design of the learning environment in use. This work aims to identify and describe the underlying user interface design principles as well as components and mechanisms for collaborative video-based learning. The objective this work is to elaborate design principles in terms of design patterns as well as computer-supported learning scripts dedicated to audiovisual media.

In order to identify and elaborate interaction design patterns, 121 video-based learning environments was examined in a content analysis with 50 different coding categories. Furthermore, patterns from related pattern languages and experiences from the development of software prototypes and conducted field studies were considered. The resulting 45 interaction design patterns represent common solutions for reoccurring problems in the design and development of video-based learning environments. The patterns were discussed at pattern conferences and evaluated from multiple perspectives. They have been used to draft, develop, and evaluate multiple video-based learning environments. The presented hypervideo framework *VI-TWO* helps to transfer the patterns into usable web components.

In the second part of the work, components and mechanisms of video-based CSCL scripts could be derived from existing video-based and general CSCL scripts. In contrast to general scripts, video-based scripts include temporal anchored tasks and video-related activities. In this perspective videos are considered as continuous media elements that can be augmented by additional contents and time-related learning activities. Three different types of video-based CSCL scripts could be identified: 1) scripts that aim to analyse and discuss the video content, 2) peer annotation and peer assessment scripts, and 3) Jigsaw scripts that enable problem-based learning. The intrinsic complexity is a distinctive feature of these scripts. The complexity is arranged at three levels: The first level is determined by timely detached phases of learning activities. If additional assignments are processed within groups (intra-group) the scripts' complexity is on level 2. The third level is characterised by assignments and achievements that involve learners across different groups (inter-group). Based on the specified script components (e.g. groups, resources, activities, system features) and mechanisms (sequencing, group formation, task anchoring), a specific data model and an user interface concept has been developed and conveyed in a CSCL system named *VI-LAB*. The system supports the different script types and complexity levels.

The contributions made to the research field of Technology-Enhanced Learning are two-fold. First, the interaction design patterns are a valuable tool for designers, developers, and instructors who are designing or implementing video-based learning environments. Second, the realised system and the specified video-based CSCL scripts lay the foundation for applying scripts in the field and enables further research about collaborative video-based learning.

N. Seidel. *Interaction Design Patterns und CSCL-Scripts für Videolernumgebungen*. Dissertation. TU Dresden, 2018. <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-233756>.

Diese Publikation ist unter *Creative Commons – Namensnennung 3.0 Deutschland* lizenziert und darf als Ganzes oder ausschnittsweise vervielfältigt, verbreitet und öffentlich zugänglich gemacht werden, sofern dies im Text nicht anders vermerkt ist.



Qucosa-URN: <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-233756>

Autor: Niels Seidel

Gestaltung und Satz: Niels Seidel /  $\text{\LaTeX}$

Datum: 5. März 2018

## Vorwort

Die vorliegende Dissertation entstand an der Professur für Produktionswirtschaft und Informationstechnik am Internationalen Hochschulinstitut Zittau – einer Zentralen Wissenschaftlichen Einrichtung der Technischen Universität Dresden. Mein besonderer Dank gilt dem Inhaber der Professur: Prof. Dr. Thorsten Claus. Er hat mir die Möglichkeiten und die Freiräume geschaffen, mich diesem umfangreichen Thema anzunehmen. Sein außerordentliches Interesse am E-Learning förderte zudem eine stetigen Verbesserung der entwickelten Ansätze und Lösungen.

Außerdem danke ich zahlreichen Personen, die auf unterschiedliche Weise zum Gelingen der Arbeit beigetragen haben. Dazu zählen insbesondere:

- Dr. Johannes Cullmann (ehem. IHP/HWRP Sekretariat, Koblenz), Dr. Marco Leidel (ehem. TU Dresden), Dr. Steffen Niemann (UFZ Leipzig) sowie Andrea Wessler (IHP/HWRP Sekretariat, Koblenz) für die Gelegenheit zusammen das E-Learning Modul *IWRM education* zu entwickeln und zu untersuchen.
- Prof. Dr. Thomas Köhler (TU Dresden) sowie seinen Mitarbeitern Dr. Daniela Pscheida und Dr. Steffen Albrecht für die inspirierende Zusammenarbeit im E-Learning Cluster des *eScience – Forschungsnetzwerk Sachsen*.
- Prof. Dr. Eric Schoop (TU Dresden) und seinen Mitarbeitern für die sehr gute Zusammenarbeit in gemeinsamen Projekten und das konstruktive Feedback bei Vorträgen. Insbesondere danke ich seiner Mitarbeiterin Corinna Hetmank, die mir ihre Lernvideos und ihr Veranstaltungskonzept für Feldstudien zur Verfügung stellte.
- Dr. Iris Tabak, die mich zu einem Forschungsaufenthalt an die Ben Gurion Universität in Be'er Scheva eingeladen und dort mit Dr. Jakov Gal und Avi Segal vom *Human-Computer Decicion-Making Lab* bekannt gemacht hat.
- Die *Shepherds* der *EuroPLoP* 2014 und 2015: Andreas Fieser, Dr. Andreas Faßbender (TU Darmstadt), Dr. Christopher Preschern, Prof. Dr. Christian Kohls (TH Köln) und Christian Köppe (HAN University) sowie die Teilnehmer der *Writers' Workshops*.
- Jenen Lehrenden, die den Facharbeitskreis „Videos in der Lehre“ zu einem Quell und Kristallisationspunkt meiner Forschung gemacht haben.
- Meinen Kolleginnen und Kollegen an der Professur für Produktionswirtschaft und Informationstechnik, die mir in zahlreichen Projekten anregende Diskussionen und wichtige Erkenntnisse bescherten.
- Luise Träger und Dr. Sabine Seidel für das Korrekturlesen dieser Arbeit.

Zittau, im November 2016

Niels Seidel





# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>19</b>
1.1	Motivation . . . . .	19
1.2	Herausforderungen und Forschungsfragen . . . . .	20
1.2.1	Interaction Design Patterns . . . . .	20
1.2.2	Videobasierte CSCL-Scripts . . . . .	22
1.3	Kapitelübersicht und Forschungsmethoden . . . . .	25
1.3.1	Kapitelübersicht . . . . .	25
1.3.2	Forschungsmethoden je Kapitel . . . . .	27
<b>2</b>	<b>Lernen mit Videos</b>	<b>29</b>
2.1	Terminologie, Einsatzfelder und Potentiale von Lernvideos . . . . .	30
2.1.1	Begriffsbestimmung . . . . .	30
2.1.2	Einsatzfelder und Szenarien für das Lernen mit Videos . . . . .	32
2.1.3	Potentiale des Medium Video . . . . .	34
2.2	Videos im Kontext kognitiver Lerntheorien . . . . .	36
2.2.1	Theorie der kognitiven Last . . . . .	36
2.2.2	Kognitive Theorie des Lernens mit Multimedia . . . . .	38
2.3	Interaktivität audiovisueller Lernmedien . . . . .	44
2.4	Lernformen . . . . .	48
2.4.1	Rezeptives Lernen . . . . .	49
2.4.2	Selbstgesteuertes Lernen . . . . .	50
2.4.3	Kollaboratives Lernen . . . . .	52
2.5	Zusammenfassung . . . . .	56
<b>3</b>	<b>Videolernumgebungen und Hypervideos</b>	<b>59</b>
3.1	Terminologie und Modelle . . . . .	60
3.1.1	Videolernumgebung . . . . .	60

3.1.2	Terminologie: Video, Hypervideo und interaktives Video . . . . .	61
3.1.3	Spezifikationen für Hypervideo-Dokumente . . . . .	65
3.1.4	Modelle des zeitlichen Layouts . . . . .	66
3.2	Human Video Interface . . . . .	68
3.2.1	Gestaltungsraum von Hypervideos . . . . .	70
3.2.2	Usability-Herausforderungen von Human Video Interfaces . . . . .	74
3.3	Technische Herausforderungen . . . . .	76
3.3.1	Download und Cache-Management / Echte Nicht-Linearität . . . . .	77
3.3.2	Synchronisierung von Video und Annotationen . . . . .	78
3.3.3	Adressierung und Abruf von Medienfragmenten . . . . .	78
3.3.4	Deklarative Ansätze der Repräsentation von Multimedia . . . . .	80
3.4	Produktion und Integration in Lernumgebungen . . . . .	81
3.4.1	Produktion: Vorgehensweisen und Paradigmen . . . . .	82
3.4.2	Integration in Lernumgebungen und Zusammenspiel mit Diensten im WWW . . . . .	85
3.5	Zusammenfassung . . . . .	87
<b>4</b>	<b>Interaction Design Patterns für Videolernumgebungen</b>	<b>91</b>
4.1	Einführung in Design Patterns . . . . .	92
4.1.1	Design Patterns . . . . .	95
4.1.2	Mustersprache . . . . .	101
4.1.3	Verwandte Ansätze im Interaction Design . . . . .	104
4.1.4	Verwandte Pattern Languages . . . . .	106
4.2	Systematische Elaboration von Design Patterns . . . . .	109
4.2.1	Stand der Forschung bzgl. dem Pattern Mining . . . . .	110
4.2.2	Inhaltsanalyse von Videolernumgebungen . . . . .	112
4.2.3	Analyse und Integration ähnlicher Muster bzw. Mustersprachen . . . . .	128
4.2.4	Verfassen sowie Revision und Evaluation der Muster . . . . .	130
4.2.5	Konstruktion der Pattern Language . . . . .	135
4.3	Pattern Language für Videolernumgebungen . . . . .	140
4.3.1	Struktur der Pattern Language . . . . .	140
4.3.2	Angrenzende Mustersprachen . . . . .	144
4.3.3	Repräsentation in einer Wissensbasis . . . . .	145
4.3.4	Anwendungs- und Einsatzszenarien . . . . .	148

4.3.5	Exemplarische Interaction Design Patterns . . . . .	151
4.4	Zusammenfassung . . . . .	168
<b>5</b>	<b>Videobasierte CSCL-Scripts</b>	<b>171</b>
5.1	Einführung . . . . .	172
5.1.1	Hintergrund zu Scripts und CSCL-Scripts . . . . .	172
5.1.2	Definition videobasierter CSCL-Scripts . . . . .	175
5.1.3	Mehrwert und Potentiale . . . . .	177
5.1.4	Typisierung videobasierter CSCL-Scripts . . . . .	178
5.2	Spezifikation videobasierter CSCL-Scripts . . . . .	184
5.2.1	Script-Komponenten . . . . .	185
5.2.2	Script-Mechanismen . . . . .	194
5.3	Modellierung von CSCL-Scripts . . . . .	197
5.3.1	Komplexitätslevel . . . . .	200
5.3.2	Verwandte Systeme und Ansätze zur Modellierung von Scripts . . . . .	201
5.3.3	Konzept für eine formale Repräsentation . . . . .	206
5.3.4	Konzept zur Modellierung im User Interface . . . . .	209
5.4	Zusammenfassung . . . . .	212
<b>6</b>	<b>Realisierung von Patterns und Scripts</b>	<b>215</b>
6.1	VI-TWO: JavaScript Framework für interaktive Videos . . . . .	216
6.1.1	Anforderungen . . . . .	217
6.1.2	Verwandte Arbeiten . . . . .	219
6.1.3	Architektur von VI-TWO . . . . .	222
6.1.4	Videoplayer . . . . .	224
6.1.5	Videoannotationen . . . . .	225
6.1.6	Makrointeraktivität in Kollektionen von Videos . . . . .	229
6.1.7	Autorenwerkzeuge . . . . .	232
6.2	VI-LAB: CSCL-System für videobasierte CSCL-Scripts . . . . .	235
6.2.1	Anforderungen . . . . .	236
6.2.2	Architektur von VI-LAB . . . . .	238
6.2.3	Modellierung videobasierter CSCL-Scripts . . . . .	241
6.2.4	Monitoring . . . . .	244
6.3	Anwendungsbeispiele für VI-TWO und VI-LAB . . . . .	246

6.3.1	Vi-Wiki . . . . .	246
6.3.2	IWRM education . . . . .	247
6.3.3	VI-LAB (Version 1) auf Basis von Wordpress . . . . .	247
6.3.4	VI-LAB (Version 2) auf Basis von node.js . . . . .	248
6.3.5	Theresienstadt explained . . . . .	249
6.4	Zusammenfassung . . . . .	252
<b>7</b>	<b>Schlussbetrachtung</b>	<b>255</b>
7.1	Beitrag der Arbeit zur Forschung . . . . .	255
7.2	Kritische Würdigung . . . . .	256
7.3	Ausblick . . . . .	257
<b>A</b>	<b>Anhang</b>	<b>259</b>
A.1	Übersicht wissenschaftlicher Konferenzen und Sonderausgaben von Journalen zum Lernen mit Videos . . . . .	260
A.2	Erweiterte Usability Guidelines für interaktive Videos . . . . .	261
A.3	Unterstützung gängiger Video-Codecs durch verbreitete Web-Browser . . . . .	267
A.4	Recherchedatenbanken und -verzeichnisse . . . . .	268
A.5	Verzeichnis der untersuchten Videosysteme . . . . .	269
A.6	Kodierschema der betrachteten Videosysteme . . . . .	287
A.7	Interrater-Reliabilität . . . . .	288
A.8	Ergebnisse der Inhaltsanalyse . . . . .	291
A.9	Ergebnisse der Analyse ähnlicher Muster . . . . .	297
A.10	Ergebnisse der Pattern-Evaluation . . . . .	307
A.11	Kategorisierung der Patterns nach dem Universal Model of User Interface . . . . .	308
A.12	Entscheidungsmodell zur Auswahl der Design Patterns . . . . .	310
A.13	Funktionale Analyse der Videolernumgebung eKomMed mit Hilfe der Design Pat- terns . . . . .	311
A.14	PLML DTD und Mongoose Schema . . . . .	318
A.15	Kurzfassung der 45 Interaction Design Patterns für Videolernumgebungen . . . . .	321
A.16	Beispiele für Scripts/CSCL-Scripts . . . . .	335
A.17	VI-TWO Codebeispiele . . . . .	341
A.18	Spezifikation von Hypervideo-Dokumenten in VI-TWO . . . . .	343
A.19	Relevante Publikationen im Kontext dieser Arbeit . . . . .	346

<b>Literaturverzeichnis</b>	<b>349</b>
<b>Abbildungsverzeichnis</b>	<b>369</b>
<b>Tabellenverzeichnis</b>	<b>371</b>
<b>Auflistungsverzeichnis</b>	<b>375</b>
<b>Verzeichnis der Algorithmen</b>	<b>377</b>
<b>Abkürzungsverzeichnis</b>	<b>379</b>



# Kapitel 1

## Einführung

### Motivation

Das Lernen mit Videos hat in den letzten Jahren an Bedeutung gewonnen. Dies liegt nicht allein an der kostengünstigen und relativ einfachen Produktion, Speicherung und Distribution von Videos, sondern auch an der Verfügbarkeit umfangreicher Lernressourcen dieser Art. Abgesehen von den umfangreichen informellen Lernangeboten finden Videos auch in formellen Online- und Blended-Learning-Angeboten an Schulen und Hochschulen eine breite Anwendung. An Hochschulen sind technische Lösungen zur Aufzeichnung und Verbreitung von Lehrveranstaltungen (z.B. Kandzia et al. [2011]; Rohs und Streule [2009]), aber auch videobasierte Lehr- und Lernformate wie *flipped classroom* (z.B. Evseeva und Solozhenko [2015]; Kim et al. [2014]; See und Conry [2014]) oder *Massive Open Online Courses* (MOOCs) [Giannakos et al. 2014] inzwischen weit verbreitet. Im Rahmen der Studierendenbefragung des CHE Hochschulranking 2015 gaben 75 % der insgesamt 27.473 Befragten an, Videos für das eigenen Studium zu nutzen oder genutzt zu haben [Persike und Friedrich 2016].

Für die Nutzung des Mediums bedarf es zumindest einen Player als Abspielkomponente der Videos. Fast immer sind die Player Bestandteil einer digitalen Lernumgebung. Je nach Ausstattung und Funktionalität ergeben sich unterschiedliche Möglichkeiten damit zu lernen oder anderen das Lernen zu ermöglichen. Web-Technologien haben dahingehen eine ganze Reihe neuer didaktischer Nutzungsformen eröffnet, die über die Distribution und Wiedergabe von audiovisuellen Lerninhalten hinausgehen. Videos sind im Zuge dieser Entwicklungen nicht mehr als abgeschlossene Lernressourcen anzusehen. Sie integrieren andere Medieninhalte und sind selbst Teil multimedialer Darstellungsformen. Anwendern stehen Werkzeuge zur Verfügung, um Videoinhalte zu modifizieren, zu ergänzen oder mit anderen Ressourcen zu verknüpfen. Nicht zuletzt handelt es sich um soziale Medien, in denen Kommunikations- und Diskussionsprozesse entlang der Zeitachse ablaufen. Im Vergleich zu anderen *Social Media* Anwendungen spielt die zeitliche Dimension der Informationsdarbietung sowie der sozialen und systembedingten Interaktion eine zentrale Rolle.

Eine Vielzahl von videobasierten E-Learning Anwendungen geben Anlass dazu, Videolernumgebungen deshalb als einen eigenständigen Typus von Lernumgebung zu betrachten. Es handelt sich dabei um eine digitale Lernumgebung, in der Videos die primäre Lernressource darstellen,

in der Lernende bei der Nutzung der Lernvideos eine technische Unterstützung erfahren und Autorenwerkzeuge für die Bearbeitung und Manipulation der Inhalte bereitstehen.

Es ist Anliegen dieser Arbeit den Gegenstand der Videolernumgebungen aus Sicht der Informatik zu beleuchten und mit den in dieser Disziplin üblichen Methoden zu untersuchen. Zum einen wird dazu das *User Interface* als Schnittstelle zwischen dem Lernenden und der Videolernumgebungen analysiert, um daraus Gestaltungsprinzipien in Form von Interaction Design Patterns zu abstrahieren. Zum anderen werden komplexe videobasierte Lernszenarien mit Hilfe CSCL-Scripts beschrieben, spezifiziert und für die Anwendung im Blended und E-Learning in einem Software System implementiert.

### Herausforderungen und Forschungsfragen

Die Entwicklung von Videolernumgebungen stellt für Lehrende, Interaction Designer und Entwickler eine herausfordernde Aufgabe dar. Eine große Herausforderung besteht in der angemessenen Anwendung interaktiver Komponenten, die den Informationszugang und die Strukturierung sowie die Organisation und Wissensanreicherung videografischer Lernressourcen fördern. Insbesondere das Arrangement von strukturierten Gruppenlernprozessen anhand von Videos war bislang nicht spezifiziert und mangels adäquater Werkzeuge auch noch nicht realisierbar.

Um diese Situation zu verbessern, sollen im Ergebnis dieser Arbeit generische Hilfsmittel und Werkzeuge entwickelt werden, die Lehrende, Interaction Designer und Entwickler bei der Konzeption und Entwicklung von Videolernumgebungen unterstützen. Dabei werden vielfältige Anwendungsmöglichkeiten gemäß individueller Anforderungen an formelle und informelle Lernarrangements in individuellen oder kollaborativen Szenarien berücksichtigt. Diese technisch-gestalterischen, formalen und softwareseitigen Lösungen sollen die Voraussetzung für die weitere Erforschung videogestützter Lernprozesse schaffen.

#### 1.2.1 Interaction Design Patterns

Warum stellt die Entwicklung von Videolernumgebungen eine Herausforderung dar? Wer eine videobasierte Lernumgebung und dabei speziell die grafische Benutzerschnittstelle konzipieren und realisieren möchte, benötigt Kenntnisse aus einer Reihe angrenzender Disziplinen. Da es sich hierbei zunächst einmal um eine E-Learning-Anwendung handelt, sind Kenntnisse in diesem Bereich unbedingt erforderlich. Für eine lernförderliche Einbindung der Videos gilt es, kognitive Theorien zum Lernen mit Multimedia zu berücksichtigen. Um bei den Anwendern eine hohe Akzeptanz zu erzeugen, sind Erfahrungen im Bereich der Mensch Computer Interaktion (MCI) mit Bezug zur Usability und der *User Experience* (UX) unentbehrlich. Da der Umgang mit Videos über klassische Abspielkomponenten eines Player hinaus geht, sind Konzepte interaktiver Videos bzw. Hypervideos zu integrieren. Know-How im Bereich Hypermedia, als die interaktive Form von Multimedia, sind dafür grundlegend. Bis hierhin wurden noch nicht einmal Aspekte der technischen Implementierung und plattformspezifischen Realisation erwähnt. Es wird jedoch bereits deutlich, dass die Gestaltung von Videolernumgebungen eine disziplinübergreifende Sicht- und Herangehensweise erfordert.



Zu klären ist, welche Richtlinien und Empfehlungen in den genannten Disziplinen das Lernen mit Videos in dafür ausgelegten Lernumgebungen adressieren.

Gestaltungsrichtlinien aus dem *User Interaction Design* (z.B. Deutsches Institut für Normung (Hsg.) [2003, 2006]), wie auch allgemeine *Interaction Design Patterns* (z.B. Tidwell [2011], van Welie und Veer [2003], van Duyne et al. [2006] oder Yahoo! [2012]) berücksichtigen kontinuierliche Medien, insbesondere im Kontext des Lernens, nur unzureichend oder überhaupt nicht.

In thematisch ähnlichen Domänen im Bereich Mensch Computer Interaktion, wie dem interaktiven Fernsehen, zeitbasierter *Hypermedia* oder bei der Produktion von *Screencasts* geben *Interaction Design Patterns* als Beschreibung guter Gestaltungslösungen eine Orientierung für die Konzeption und Entwicklung solcher Systeme (z.B. Chen und Rabb [2010]; Kunert [2009]; Lopes und Carriico [2007]). Mediendidaktische Aspekte wurden in letztgenannten Arbeiten, wenn überhaupt, nur marginal berücksichtigt. Videolernumgebungen unterscheiden sich durch ihren Anwendungskontext und ihre Ausführungsumgebung somit deutlich von technischen Systemen im Bereich des Entertainments und des interaktiven Fernsehens.

Auch im Bereich des E-Learning gibt es bereits einige Vorarbeiten, die auf Patterns aufbauen und auf diese Weise erfolgreiche Lösungen und Praktiken des computer- und internetgestützten Lernens und Lehrens vermitteln (vgl. Überblick bei Kohls [2014]). Von einer Ausnahme eines einzelnen Patterns [Kortenkamp und Blessing 2011] abgesehen, wird dem Lernen mit Videos dabei kein besonderer Stellenwert beigemessen. Videos gelten vielfach als in sich abgeschlossene Broadcast-Medien, ohne einen Rückkanal oder Möglichkeiten zur Interaktion mit dem Video oder anderen Anwendern.

Wichtige Impulse und evidenzbasierte Erklärungen zur Wirkung und Wahrnehmung von Lernvideos finden sich jedoch in den Prinzipien des Lernens mit Multimedia von Mayer [2009] sowie in den Theorien und Prinzipien zur Vermeidung kognitiver Last von Clark et al. [2006]. Es werden dabei jedoch nur einzelne Teilaspekte von Lernumgebungen betrachtet, während technische Gegebenheiten interaktiver Videos im Kontext des Web 2.0 sowie die Interaktionen der beteiligten Personen keine Beachtung finden.

Im Bereich *Hypermedia* finden sich in der Literatur nur vereinzelte Guidelines zu speziellen Aspekten, wie DVD-Menüs [Költringer et al. 2005], der Usability von interaktiven Videos [Meixner et al. 2011] oder der *Multimedia-Reuse Pattern Language* von Cybulski und Linden [1998].

In den genannten Disziplinen werden die Teilaspekte der Gestaltung von Videolernumgebungen nur bruchstückhaft thematisiert. Die Erarbeitung eines ganzheitlichen Ansatzes, in dem unterschiedliche fachliche Sichtweisen auf der Gestaltung der Benutzerschnittstelle einer Videolernumgebung integriert sind, stellt deshalb eine Forschungslücke dar.

Ausgangspunkt ist dabei die Schnittstelle zwischen Mensch und Video, die sich durch die Form und Funktion einer Vielzahl von Elementen auszeichnet. Viele dieser Elemente stellen als Gestaltungslösung keine singulären Erscheinungen, sondern wiederkehrende Lösungen mit verschiedenen Ausprägungen dar. Anstatt die Frage nach der Gestaltung von Videolernumgebungen mit einer einzigen beispielhaften Lösung zu beantworten, werden einzelne Elemente als bewährte Teillösungen für wiederkehrende (Teil-)Probleme im jeweiligen Anwendungskontext beschrieben. Diese Lösungen für ein Problem im Kontext bezeichnet man als Entwurfsmuster (engl. *Design Pattern*), oder kurz als Muster oder Pattern. Das charakteristische Beschreibungs-

format von Patterns erlaubt es, zum einen verschiedene fachliche Perspektiven zu integrieren und zum anderen Widersprüche sowie Vor- und Nachteile darin zu diskutieren. Die Forschungsfrage lautet deshalb:

***Forschungsfrage FF 1: Welche bewährten Lösungen für häufig auftretende Probleme (Design Patterns) gibt es hinsichtlich der Gestaltung der grafischen Benutzerschnittstelle von Videolernumgebungen?***

Die Zielsetzung besteht darin, wiederkehrende Lösungen für verbreitete Probleme bei der Gestaltung und Entwicklung von Videolernumgebungen zu identifizieren, als Interaction Design Patterns zu beschreiben, und abschließend in einer Pattern Language zusammenzufassen.

Abgesehen von der Konzeption und Gestaltung von Videolernumgebungen stellt auch die Implementierung entsprechender Anwendungen eine Hürde dar. Die Herausforderung ergibt sich aus dem Umstand, dass videografische Lernressourcen für den Anwender schwerer zu handhaben sind. Videos lassen sich aufgrund ihrer zeitlichen Struktur und Informationsdarbietung nicht ohne Weiteres wie Texte oder statische Bilder bearbeiten, anpassen oder durchsuchen. Dies gilt insbesondere dann, wenn eine große Anzahl von Videos vorliegt. Derzeit fehlt es an umfassenden und flexibel einsetzbaren Software Bibliotheken oder Frameworks für webbasierte Hypervideo-Player, die sich insbesondere in Web 2.0 Anwendungen (einschließlich Lernumgebungen) integrieren lassen. Hierbei handelt es sich um eine weitere Forschungslücke. Um die Realisierung von Videolernumgebungen zu erleichtern und die Patterns auch in der Praxis erproben zu können, besteht das zweite Ziel darin, für diese Zwecke ein leistungsfähiges und generisches Software Framework für interaktive Videos zu entwickeln.

### 1.2.2 Videobasierte CSCL-Scripts

Die didaktische Gestaltung von Lernszenarien umfasst neben den Lernzielen, -aufgaben und -ressourcen vor allem die Interaktionen der Beteiligten in Abhängigkeit organisatorischer und technischer Rahmenbedingungen. Um solche Szenarien im E- oder Blended Learning einsetzen zu können, muss man sie in entsprechenden Lernumgebungen abbilden. Die auf Schank und Abelson [1977] zurückgehende Script-Theorie bietet dazu einen umfassenden Rahmen. Im Kontext des *Computer Supported Collaborative Learning* (CSCL) werden *CSCL-Scripts* insbesondere dazu eingesetzt, kollaborative Lernprozesse in dafür ausgelegten Systemen zu strukturieren und für die Durchführung von Lehrveranstaltungen zu modellieren. Dies geschieht durch die Sequenzierung von Aktivitäten und durch Rollenfestlegung innerhalb von Gruppen sowie durch bestimmte Interaktionen zwischen den beteiligten Lernenden [Dillenbourg und Tchounikine 2007]. Dadurch lassen sich sowohl individuelle, kooperative und kollaborative, als auch kollektive Lernaktivitäten definieren und miteinander kombinieren [Weinberger und Fischer 2012]. Aus organisatorischer Sicht kann sich dadurch der Aufwand für Lehrende verringern, onlinebasierte Gruppenlernszenarien auch für eine größere Anzahl von Teilnehmern zu realisieren.

CSCL-Scripts wurden bislang hauptsächlich für die Arbeit mit und an Texten eingesetzt und untersucht (z.B. Aronson et al. [1978]; Berger et al. [2001]; Dillenbourg [2006]; Engeli [2001]; Johnson et al. [2006]; Mandl und Ertl [2004]; O'Donnell und Dansereau [1993]; Weinberger et al. [2005]). Im Kontext des videobasierten Lernens gibt es weitaus weniger Literatur [Lauer

und Trahasch 2005a; Trahasch 2006]. Einige Autoren beschreiben Scripts ohne explizite Verwendung des Begriffs [Leijen et al. 2009; Pea und Lindgren 2008; Shirouzu und Miyake 2007; Vohle 2013]. Zur Umsetzung dieser Scripts wurden solitäre Anwendungen eingesetzt, mit denen jeweils nur der betreffende Typ von Script konfiguriert und umgesetzt werden konnte. Es fehlt demzufolge an einem System, welches generische CSCL-Scripts auf Basis von Videos realisieren kann. Um ein solches, universelles CSCL-System jedoch realisieren zu können, sind Erkenntnisse über die Elemente und Strukturen der betreffenden Scripts erforderlich, um daraus eine formale Repräsentation der Daten und der Benutzerschnittstelle zur Modellierung der Scripts zu entwickeln. Aus didaktischer Sicht gilt es dabei auch mögliche Formen der Zusammenarbeit von Lernenden innerhalb von Kleingruppen zu berücksichtigen. Gegenwärtig werden Videos vorwiegend im Sinne des rezeptiven und bestenfalls selbstgesteuerten Lernens eingesetzt. Das kollaborative Lernen anhand von Videos wird nur von sehr wenigen Systemen und dann auch nur in Bezug auf sehr wenige Aktivitäten unterstützt. Zusammengefasst ergibt sich darauf folgende Forschungsfrage:

***Forschungsfrage FF 2: Wie lassen sich videobasierte Lernprozesse mit Hilfe von CSCL-Scripts innerhalb von Lernumgebungen realisieren?***

Die Beantwortung dieser Fragestellung umfasst drei Teilfragen, welche die Spezifikation, Formalisierung und Modellierung videobasierter CSCL-Scripts betreffen und auf die im Folgenden genauer eingegangen wird.

In den von Kobbe et al. [2007] analysierten und spezifizierten CSCL-Scripts finden Aspekte des videobasierten Lernens keine ausreichende Berücksichtigung. Eine Reihe von Vertretern videobasierter Scripts (z.B. Seidel [2013]; Trahasch [2006]; Vohle [2013]) lassen sich damit nicht beschreiben. Insbesondere finden videobezogene Lernaktivitäten und die dafür notwendigen softwareseitigen Voraussetzungen keine Berücksichtigung. Aus diesem Grund ist eine umfassender Analyse erforderlich, die erstens den Besonderheiten kontinuierlicher Medien, zweitens den Möglichkeiten interaktiver Videos und drittens spezifischen Lernaktivitäten und -aufgaben Rechnung trägt. Die Forschungsfrage lautet deshalb:

***Forschungsfrage FF 2a: Mit Hilfe welcher Komponenten und Mechanismen lassen sich videobasierte CSCL-Scripts spezifizieren?***

Ausgehend von den derzeit bekannten Vertretern videobasierter CSCL-Scripts und unter Berücksichtigung von Beispielen allgemeiner CSCL-Scripts sollen zunächst Komponenten videobasierter Scripts identifiziert und beschrieben werden. Darauf aufbauend sind Mechanismen zu charakterisieren, die die Komponenten miteinander in Beziehung setzen und den Ablauf eines Scripts gewährleisten. Als Ergebnis der Analyse soll eine Spezifikation für weitgehend generische videobasierte CSCL-Scripts entstehen. Diese Spezifikation stellt die Grundlage für eine formale Datenspezifikation dar und definiert gleichzeitig die Anforderungen für ein CSCL-System zur Modellierung und Anwendung dieser Scripts.

CSCL-Scripts im Allgemeinen und videobasierte Scripts im Speziellen sind auf die Ausführung in dafür ausgelegten Lernumgebungen beschränkt. Um videobasierte Scripts in CSCL-Systemen implementieren zu können, ist es sinnvoll eine formale Repräsentation zu spezifi-

zieren und diese vom ausführenden System zu trennen [Lauer und Trahasch 2005a]. Eine Formalisierung der möglichen Ausprägungen videobasierter CSCL-Scripts erlaubt es somit, Scripts innerhalb verschiedener Lernumgebungen wiederzuverwenden und interoperabel einzusetzen. Bislang existieren jedoch nur statische Lösungen, mit denen in einem System jeweils nur eine Art von Script ausgeführt und konfiguriert werden kann (z.B. Reinhardt et al. [2011]; Shirouzu und Miyake [2007]; Trahasch [2006]; Vohle [2013]). Eine formale Beschreibung für generische Scripts ist darüberhinaus eine Voraussetzung, um neuartige Scripts für Forschung und Lehre modellieren, weitergeben und gesondert untersuchen zu können.

Bestehende Ansätze zur formalen Repräsentation von allgemeinen, d.h. nicht zwingend videobasierten Scripts berücksichtigen entweder keine Videos (z.B. [Harrer et al. 2007; Höver und Mühlhäuser 2014; Miao et al. 2005]) oder sie berücksichtigen nur einzelne Interaktionen mit dem Medium, wie etwa zeitgenaue Kommentare [Trahasch 2006]. Folgende Forschungsfrage adressiert diese Forschungslücke:

***Forschungsfrage FF 2b: Wie lassen sich generische videobasierte CSCL-Scripts formal beschreiben?***

Zur Beantwortung dieser Frage sollen bestehende CSCL-Systeme und formale Beschreibungen für allgemeine Scripts hinsichtlich ihrer Eignung für videobasierte CSCL-Scripts untersucht werden. Dabei besteht die Anforderung, generische Scripts in Anbetracht ihrer Spezifikation vollständig und unabhängig von der Anzahl der darin repräsentierten Entitäten (z.B. Videos, Gruppen) abzubilden. Des Weiteren sollte eine formale Beschreibung anpassbar sein und im Idealfall auf bestehenden Standards aufsetzen bzw. diese in geeigneter Weise ergänzen, um eine Interoperabilität zu gewährleisten.

Ausgehend von einer formalen Repräsentation für videobasierte Scripts ist nun abschließend zu klären, wie man Endanwender in die Lage versetzt Scripts selbst zu modellieren oder anzupassen. In Anbetracht der verschiedenen Einflussgrößen (Script Komponenten) und deren Abhängigkeiten (Script Mechanismen) können sehr komplexe Scripts entstehen, die nicht mehr ohne adäquate Hilfsmittel überblickt und modifiziert werden können,

Für die Modellierung von Scripts durch Endanwender gibt es bereits eine Reihe von Lösungsansätzen. Ein Ansatz bilden Auszeichnungssprachen wie z.B. die XML-Schemata von *IMS LD*, die mit Hilfe eines Texteditors erstellt und bearbeitet werden können. Anwender müssen dazu mit den entsprechenden Ausdrücken der Notation vertraut sein und diese syntaktisch korrekt anwenden. Grafische Bedienkonzepte stellen einen alternativen Ansatz dar. Sie umfassen beispielsweise Flussdiagramme (z.B. in *Collage 2*), Workflow-Ansichten (z.B. in *COW*), Modellierungssprache (z.B. *S-BPM*) und *Petri-Netze* sowie grafische Programmierkonzepte (z.B. *blockly* oder *Scratch*). An Endanwender werden dabei gewisse Erwartungen gestellt. Entweder müssen sie sich mit einer Syntax oder grafischen Notation vertraut machen, oder sich in ein ungewohntes Bedienkonzept einarbeiten. Alle genannten Ansätze weichen somit von den in Lernsoftware üblichen Bedienkonzepten sehr stark ab. Insbesondere die Auszeichnungs-, Modellierungs- und Programmiersprachen erfordern vom Anwender vertiefte Kenntnisse und sind für Laien nicht leicht zu erlernen. Hinzu kommt, dass die bestehenden Ansätze videobasierte CSCL-Scripts nur unvollständig abbilden. Neben der Erlernbarkeit und Vollständigkeit stellt

die Komplexität des Scripts eine weitere Herausforderung für den Nutzer dar. Komplexität kann einerseits durch eine hohe Anzahl an Entitäten wie Videos oder Gruppen entstehen oder auf deren komplizierte Verflechtung, etwa durch wechselnde Gruppenformationen, Zugriffsrechte oder erlaubte Interaktionen, bedingt sein. Wie kompliziert ein videobasiertes CSCL-Script jedoch aufgebaut sein soll, liegt im Ermessen der Lehrenden als Anwender. In dem für die Modellierung eingesetzten System müssen jedoch Vorkehrungen getroffen werden, die Bedienbarkeit unabhängig von der Komplexität des Scripts gewährleisten zu können. Es stellt sich deshalb die Frage:

***Forschungsfrage FF 2c: Wie lassen sich videobasierte CSCL-Scripts im User Interface einer Videolernumgebung modellieren?***

Das Ziel besteht zunächst darin ein Bedienkonzept zu entwickeln, welches einerseits die formalen Bestandteile generischer Scripts repräsentiert und andererseits die mögliche Komplexität eines Scripts gegenüber dem Anwender auf ein notwendiges Maß reduziert. Anwender sollen dabei alle erforderlichen Daten und Prozesse auch ohne Kenntnis von Auszeichnungs- oder Programmiersprachen in einer grafischen Bedienoberfläche konfigurieren und manipulieren können. Auf Grundlage dieses Bedienkonzepts soll ein prototypisches CSCL-System entwickelt werden, in welchem das Bedienkonzept zur Modellierung von videobasierten CSCL-Scripts für die praktische Anwendung derselben realisiert ist. Im Ergebnis soll es Anwendern möglich sein, videobasierte CSCL-Scripts gemäß ihren eigenen Anforderungen und Vorstellungen zu modellieren und in der Lehre einzusetzen.

## Kapitelübersicht und Forschungsmethoden

### 1.3.1 Kapitelübersicht

Das vorliegende Buch umfasst sieben Kapitel. Im Folgenden werden die Themen und Inhalte der nachfolgenden Kapitel kurz beschrieben.

**Kapitel 2: Lernen mit Videos** Im ersten einführenden Kapitel steht das Lernen mit Videos im Kontext des E-Learnings im Mittelpunkt. Vor dem Hintergrund typischer Einsatzfelder für Videos werden Potentiale dieses Mediums diskutiert und mit den aus den kognitiven Lerntheorien bekannten Gestaltungsprinzipien in Beziehung gebracht. Gesondert betrachtet werden die Interaktivität mit dem System und die Interaktionen zwischen den beteiligten Lernenden. Beide Aspekte finden sich in den jeweiligen Lernformen wieder. Im Zusammenhang mit dem Medium Video und den dafür ausgelegten Videolernumgebungen wird auf das rezeptive, selbstgesteuerte und kollaborative Lernen genauer eingegangen.

**Kapitel 3: Videolernumgebungen und Hypervideos** Das zweite einführende Kapitel beleuchtet technische Aspekte von Videolernumgebungen und Hypervideos. Ausgehend von Begriffsdefinitionen und grundlegenden Modellen werden Herausforderungen für die Realisierung solcher Systeme im WWW erläutert. Desweiteren geht es um die Schnittstelle zwischen Mensch und Video („human video interface“) und die sich daraus ergebenden Schwierigkeiten für die Usability entsprechender Anwendungen. Ein Teilaspekt davon, die Erstellung von Hypervideos, wird genauer betrachtet.

**Kapitel 4: Interaction Design Patterns für Videolernumgebungen** Ziel dieses Kapitels ist es, bewährte Gestaltungslösungen des *User Interfaces* von Videolernumgebungen aus einer Vielzahl von Einzelfällen zu identifizieren und ihre Notwendigkeit durch Angabe der dafür ursächlichen Probleme im jeweiligen technischen oder didaktischen Kontext zu begründen. Nach einer grundlegenden Einführung in die Mustertheorie wird eine systematische Vorgehensweise vorgestellt, um wiederkehrende Gestaltungslösungen im *User Interface Design* von Videolernumgebungen zu identifizieren und daraus *Interaction Design Patterns* sowie eine *Pattern Language* zu entwickeln. Als Ergebnis der Methodenanwendung steht eine Pattern Language mit 45 Patterns, welche hinsichtlich ihrer Repräsentationsformen und Einsatzmöglichkeiten erläutert wird.

**Kapitel 5: Videobasierte CSCL-Scripts** In diesem Kapitel werden videobasierte CSCL-Scripts begrifflich eingeführt und von allgemeineren Scripts abgegrenzt. Anhand von Beispielen aus der Literatur werden anschließend drei grundlegende Formen sowie Merkmale videobasierter Scripts erläutert. Ausgehend von diesen Beispielen und einer Analyse verschiedener Scripts erfolgt eine Spezifikation von Komponenten und Mechanismen. Aus dieser Spezifikation wird anschließend eine formale Repräsentation der für die Ausführung eines Scripts notwendigen Daten entwickelt sowie ein Bedienkonzept entworfen, welches Endanwender in die Lage versetzt, Scripts nach ihren eigenen Anforderungen zu modellieren und zu konfigurieren.

**Kapitel 6: Realisierung** Mit *VI-TWO* und *VI-LAB* werden in diesem Kapitel zwei generische Werkzeuge vorgestellt, die Lehrende, Interaction Designer und Entwickler bei der Konzeption, Entwicklung und Nutzung von Videolernumgebungen unterstützen. *VI-TWO* ist ein modulares Software Framework für interaktive Videos, mit dem sich zum einen die in Kapitel 4 ermittelten *Interaction Design Patterns* in die Praxis überführen lassen und zum anderen vielfältige Videolernumgebungen entwickelt werden können. Eine solche Lernumgebung stellt das CSCL-System *VI-LAB* dar. Es ermöglicht die Modellierung und Ausführung generischer videobasierter CSCL-Scripts.

**Kapitel 7: Zusammenfassung** Im letzten Kapitel werden die Ergebnisse aus den Kapitel 2 – 6 zusammengefasst und im Zusammenhang diskutiert. Da die im Rahmen dieser Arbeit vorgestellten Werkzeuge eine ganze Reihe von Untersuchungen ermöglichen geht es im Ausblick um Anknüpfungspunkte für mögliche Weiterentwicklungen sowie empirische Untersuchungen der hier aufgezeigten videogestützten Lernformen.

**Anhang** Im Anhang dieser Arbeit finden sich wichtige Teilergebnisse und Rohdaten der in den Kapiteln 2 – 6 erwähnten Untersuchungen. Darüberhinaus sind in Anhang A.15 (S. 321) die Kurzfassungen der 45 Patterns dargestellt.



### 1.3.2 Forschungsmethoden je Kapitel

Die inhaltliche Struktur der einzelnen Kapitel orientiert sich an dem von Holz et al. [2006] entwickelten allgemeinen Rahmen für Forschungsmethoden in der Informatik. Dieser fußt auf vier Fragen:

- A: Was soll erreicht werden?
- B: Woher stammen die Daten?
- C: Was wird mit den Daten gemacht?
- D: Wurde das Ziel erreicht?

Für die Kapitel 2 – 6 wurden Forschungsziele durch Fragestellungen (A) definiert, die mit Hilfe bestimmter Daten (B) und unter Anwendung passender Methoden (C) systematisch bearbeitet und in Anbetracht des Ziels überprüft (D) wurden. Im Verlauf der Kapitel ergeben sich Iterationen, indem Ergebnisse eines Kapitels die Datengrundlage für ein anderes Kapitel bilden. Tabelle 1.1 gewährt für die einzelnen Kapitel einen Überblick über die jeweilige Zielstellung, Datengrundlage, angewendeten Methoden und die Zielerreichung. Durch die in der Tabelle enthaltenen Querverweise, stellt sich dem Leser dar, in welcher Weise die Kapitel aufeinander aufbauen.

**Tabelle 1.1:** Übersicht der Methodologie je Kapitel

	<b>Kapitel 2: Lernen mit Videos</b>	<b>Kapitel 3: Videolernumgebungen und Hypervideos</b>	<b>Kapitel 4: Interaction Design Patterns für Videolernumgebungen</b>	<b>Kapitel 5: Videobasierte CSCL-Scripts</b>	<b>Kapitel 6: Patterns und Scripts umsetzen mit VI-TWO und VI-LAB</b>
<b>A: Was soll erreicht werden?</b>	Wie kann man mit Videos lernen und welchen Gestaltungsprinzipien unterliegen sie?	Identifikation technischer Herausforderungen hinsichtlich der Realisierung von Videolernangeboten	<b>FF 1:</b> Identifikation und Beschreibung bewährter Lösungen für häufig auftretenden Probleme (Design Patterns) hinsichtlich der Gestaltung des <i>User Interfaces</i> von Videolernumgebungen	<b>FF 2</b> Ansatz zur Realisierung videobasierter Scripts in Lernumgebungen	Entwicklung generischer Software Systeme für die Umsetzung der Interaction Design Patterns und für videobasierten CSCL-Scripts
<b>B: Woher stammen die Daten?</b>	Literatur	Literatur; Systemvergleiche	Existierende Videolernumgebungen; Literatur; verwandte Interaction Design Patterns; Erfahrungen aus der Realisation von Patterns in Videolernumgebungen; Feldstudien	Literatur; videobasierte und sonstige CSCL-Scripts; Patterns aus Abschn. 4.3	Design Patterns; existierende Videolernumgebungen; Erfahrung bei der Entwicklung und von Feldstudien; Anforderungen aus Kapitel 2 und 3
<b>C: Was wird mit den Daten gemacht?</b>	Analyse von Gestaltungsprinzipien und Lernformen	Beschreibung der Schnittstelle zwischen Mensch und Video; Identifikation von technischen Herausforderungen und Usability-Problemen; Analyse von Ansätzen zur Produktion von Hypervideos	Inhaltsanalyse von Videolernumgebungen; Analyse verwandter Pattern Languages; Analyse des Nutzerverhalten	<b>FF 2a:</b> Ableitung von Spezifikationen einzelner Komponenten und Mechanismen videobasierter Scripts; <b>FF 2b:</b> formale Beschreibung der Scripts; <b>FF 2c:</b> Entwicklung eines Workflows für die Modellierung durch den Anwender	Anwendung von Design Patterns zur Implementierung eines Software Frameworks für Hypervideos; Realisierung eines CSCL-Systems für videobasierte Scripts
<b>D: Wurde das Ziel erreicht?</b>	Definition von Videolernumgebung; Prüfung von Gestaltungsprinzipien; Beschreibung von Lernformen;	Ableitung technischer und funktionaler Anforderungen	Feedback aus der Pattern Community; Anwendung der Patterns bei der Implementierung von Videolernumgebungen in Abschn. 6.2.3	Spezifizierung von CSCL-Scripts in Abschn. 5.2.1; Realisierung der Spezifikationen und Umsetzung bekannter Scripts in Abschn. 6.2	Erprobung des Frameworks in verschiedenen CMS/LMS in Abschn. 6.3; Einsatz des CSCL-Systems im Feld in Abschn. 6.3;



## Kapitel 2

# Lernen mit Videos

In den vergangenen Jahren haben Videos im *World Wide Web* (WWW) enorm an Popularität gewonnen. Laut Prognosen des Netzdienstleisters *CISCO* soll das Datenaufkommen durch *Video-On-Demand* (VOD) zwischen 2013 und 2018 jährlich um 29 % weiter ansteigen [Cisco 2014]. Die Entwicklung in diesem Bereich wird zum einen durch kostengünstige und umfangreiche Speicher- und Übertragungskapazitäten begünstigt und liegt zum anderen an der zunehmenden Verbreitung von digitalen Kameras, insbesondere in mobilen Endgeräten wie *Smartphones* oder *Tablet Computer*.

Deutlich wird dies u.a. anhand der Zahlen von *YouTube* als größten *Video Sharing Service* im WWW. Das Soziale Netzwerk von *YouTube* zählt mehr als eine Milliarde registrierte Nutzer und verfügt in Deutschland über eine größere Nutzerschaft als Facebook [Hündgen und Gugel 2013]. Des Weiteren ist *YouTube* nach *Google Search* die Nummer 2 der Suchmaschinen [Hündgen und Gugel 2013]. Nutzer laden dort monatlich 4,32 Millionen Stunden Videomaterial hoch und rezipieren Filme im Umfang von 6 Milliarden Stunden<sup>1</sup>.

Dieser Trend spiegelt sich auch im Angebot und im Gebrauch von Lernvideos wieder, und zwar in fast allen Disziplinen. Beispielsweise haben seit der Einführung von *YouTube EDU* im März 2009 mehr als 300 internationale Hochschulen, Schulen und andere Bildungseinrichtungen über 700.000 Lernvideos hinterlegt. Die Kategorie EDU zählte 2011 zu den am drittschnellsten wachsenden der Videoplattform [Glader 2012]. Zwischen den Jahren 2010 und 2011 hat sich die Zahl der Videoaufrufe auf 22 Milliarden verdoppelt [Glader 2012]. Auch in anderen *YouTube* Kategorien finden sich beträchtliche Lernressourcen. Typische Suchbegriffe wie „how to“ oder „learn to“ liefern mehr als 173,2 Millionen Treffer (Stand Februar 2015). Ursächlich für den Umfang verfügbarer videografischer Lernressourcen sind nicht allein die technischen Produktions- und Distributionsmöglichkeiten, sondern auch die didaktischen Einsatzmöglichkeiten und Potentiale des Medium Wissen zu konservieren. Ausgehend von der Definition des Begriffs Lernvideo werden zu Beginn dieses Kapitels typische Einsatzfelder und Potentiale des Mediums dargelegt. Die Potentiale wiederum sind eng mit kognitiven Prozessen bei der Verarbeitung audiovisueller Medien verbunden. Einige kognitive Lerntheorien können uns dabei helfen, diese Prozesse zu verstehen. Der zweite Teil des Kapitels beschäftigt sich deshalb mit der Fragestel-

---

<sup>1</sup>Vgl. <http://www.youtube.com/yt/press/statistics.html> (abgerufen am 15.02.2015).

lung, wie Prinzipien kognitiver Lerntheorien bei der Gestaltung von Videolernumgebungen zu berücksichtigen sind.

Neben der Rezeption ermöglichen Lernumgebungen in erheblichem Maße die Interaktion mit den Lernmedien. Welche graduellen Abstufungen die Interaktion zwischen Lernenden und System unterliegen, wird im dritten Teil des Kapitels anhand von Beispielen beschrieben. Während die technischen Aspekte interaktiver Videos erst im folgenden Kapitel in den Mittelpunkt rücken, wird an dieser Stelle der Stand der psychologisch-didaktischen Forschung zusammengefasst.

Im letzten Teil des Kapitels geht es noch einmal um die Praxis des Videoeinsatzes in der formellen Lehre wie sie vornehmlich an Hochschulen stattfindet. Kandzia et al. [2011] und Rohs und Streule [2009] gehen davon aus, dass sich die technischen Systeme zur Aufzeichnung und Distribution von Vorlesungen an deutschen Hochschulen inzwischen etabliert haben. Im Zuge dessen haben sich in den letzten Jahren neue Lehr- und Lernformate wie *Massive Open Online Courses* (MOOCs) [Giannakos et al. 2014] oder *Flipped/Inverted Classroom* [Bergmann und Sams 2012; Bishop und Verleger 2013; Hussey et al. 2014; Kim et al. 2014] entwickelt und verbreitet, die wesentlich auf videobasierte Lernressourcen aufbauen. Unabhängig vom didaktischen Arrangement finden sich in diesen Ausführungen drei unterschiedliche Lernformen wieder. Welche das sind und worin ihre Chancen und Grenzen bestehen, ist Gegenstand des letzten Teils dieses Kapitels. Im Besonderen interessiert dabei die Fragestellung, welche Gestaltungsanforderungen sich aus den Lernformen für Videolernumgebungen ergeben.

## Terminologie, Einsatzfelder und Potentiale von Lernvideos

### 2.1.1 Begriffsbestimmung

Für alle weiteren Ausführungen ist es zunächst notwendig den Begriff Video im Kontext des Lernens genauer zu bestimmen. Der Begriff des Lernvideos wird in der deutsch- und englischsprachigen Literatur häufig nur durch eine Reihe von nicht ganz trennscharfen Untergattungen benannt, z.B. *video lectures*, *d-lecture*, *e-lecture*, *web-based lecture*, *videotaped lecture*, *tele-lecture*, *digital lecture*, *lecture recording*, *presentation recording*, *video streaming*, *VideoPodcast*, *videobasierte Hyperstories*, *instructional video*, *streaming media*, *vid cast* oder *video on demand* sowie Vorlesungsaufzeichnung, Vorlesungsmitschnitt und Vortragsaufzeichnung.

Niegemann et al. [2008, S. 264] definieren Videos im Kontext des Lernens sehr eng als Bewegtbilder, die digitale oder analoge Aufnahmen der Realität darstellen. Sie verweisen aber auch darauf, dass computergenerierte Videos kaum noch von realen Aufnahmen zu unterscheiden sind. Aufgrund der Kombination von Bild und Ton kann es sich um multimediale Objekte handeln [Seel und Ifenthaler 2009, S. 35], sofern die enthaltenen visuellen und akustischen Informationen vom Betrachter wahrgenommen werden können.

Zahn [2004] subsumiert Filme, Videos und nicht-lineare audiovisuelle Hypermedien als dynamisch-visuelle Medien denen es gemein ist, „dass sie aus Bildinformationen bestehen, die sich bewegen und dass sie als Informations- bzw. Wissensmedien die Funktion haben, komplexe und dynamische Inhalte möglichst gut zu veranschaulichen und zu verdeutlichen bzw. die kognitive Verarbeitung komplexer Zusammenhänge oder Abläufe zu erleichtern.“ [Zahn 2004].

Ergänzt um den Hinweis, dass es sich bei den filmischen Medien um Aufzeichnungen und keine in Echtzeit erzeugten und übertragenen Medienströme (wie z.B. in einer Videokonferenz) handelt, soll die folgende Definition gelten:

**Definition 1.** *Lernvideos kombinieren Aufzeichnungen von Bewegtbildinformationen mit Audioinformationen, die als Informations- bzw. Wissensmedien die Funktion haben, komplexe und dynamische Inhalte möglichst gut zu veranschaulichen und zu verdeutlichen bzw. die kognitive Verarbeitung komplexer Zusammenhänge oder Abläufe zu erleichtern.*

Videorepräsentation						Aufnahmeformat
Animation						
Stop Motion	Legetechnik	Slidecast	2D / 3D	Screencast	Realität	
✓	✓	✓	✓	✓	✓	
(-)	(-)	(✓)	(✓)	(✓)	✓	Multi-angle
(-)	(-)	(-)	(✓)	-	✓	Sphärisch

**Abbildung 2.1:** Gegenüberstellung von gängigen Repräsentationen und Aufnahmeformaten von Lernvideos. Die Häkchen kennzeichnen mögliche Kombinationen.

Diese Definition schließt die in Abb. 2.1 dargestellten Videorepräsentationen und Aufnahmeformate mit ein. Bei den Repräsentationen sind grundlegend Aufnahmen realer Situationen von animierten Bewegtbildern zu unterscheiden. Screencasts bilden eine Ausnahme, da diese Aufzeichnung von einem Computerdisplay weder als real, noch als vollkommen animiert angesehen werden kann. Bei den Animationen finden sich im Bereich der Lernvideos weitere Unterkategorien:

- Stop Motion Filme: zeitlich animierte Einzelbilder.
- Legetechnikfilme (oder auch Strukturlegetechnik): ausgeschnittene, flache Objekte aus Papier oder Pappe werden mit der Hand auf eine horizontale Fläche gelegt und bewegt, während eine senkrecht darüber montierten Videokamera den Vorgang filmt.
- Slidecasts: vertonte Folge von Bildern oder Präsentationsfolien
- 2D oder 3D Animationen: sonstige Animation wie z.B. Zeichentrick oder 3D Computera-  
nimationen.

Hinsichtlich der Aufnahmeformate sind neben den Standardformaten (z.B. 4:3 und 16:9) zeitlich synchrone Aufnahmen aus mehreren Kameraperspektiven, sogenannte *Multi-angle Videos*, sowie sphärische Aufnahmen wie z.B. 360° Videos zu unterscheiden.

### 2.1.2 Einsatzfelder und Szenarien für das Lernen mit Videos

In Anlehnung an Wannemacher [2016, S. 62] gibt es drei sich gegenseitig ausschliessende Lernszenarien, in denen Videos in der digitalen Lehre eingesetzt werden. Diese unterscheiden sich primär hinsichtlich der zeitlichen und räumlichen Verteilung und dem Stellenwert der Videoressourcen.

- **Ergänzung:** Als Ergänzung oder auch Anreicherung (vgl. Wannemacher [2016]) zu Präsenzveranstaltungen oder Online-Kursen können Videomitschnitte bzw. sonstige Lernvideos unabhängig vom Lernszenario zur Verfügung gestellt werden. Lernende können diese Ressourcen zu einem beliebigen Zeitpunkt und von einem beliebigen Endgerät aus abrufen. Die Durchführung der Lehrveranstaltung ist dabei unabhängig von der Verfügbarkeit der Videos [Mertens 2007].
- **Blended Learning:** Präsenz- und Onlineanteile einer Lehrveranstaltung sind inhaltlich aufeinander abgestimmt. Die Online- und Präsenzlernphasen erfolgen im zeitlichen Wechsel oder ergänzen sich zumindest. Der Einsatz von Videos sowie anderen digitalen Lernressourcen erfolgt i.d.R. während der Onlinephasen. Dieses Konzept wird auch als hybrides Lernen oder als „Integrationskonzept“ bezeichnet. Die *Flipped Classroom* bzw. *Inverted Classroom* Methode, in der vielfach Videos genutzt werden (z.B. Evseeva und Solozhenko [2015]; Kim et al. [2014]; See und Conry [2014]), zählt ebenfalls zu diesem Szenario. [Wannemacher 2016]
- **Online-Lernen:** Digitale Lernangebote werden fast ausschliesslich Online genutzt. Die dabei eingesetzten Lernressourcen können dabei zu einem großen Teil aus Videos bestehen (z.B. bei xMOOCs [Giannakos et al. 2014]).

Darüber hinaus können unterschiedliche didaktische Intentionen mit dem Einsatz eines Videos verbunden sein [Schwartz und Hartman 2007]. Ein Video kann dazu dienen, einen Vorgang oder eine Situation zu *beobachten*, ein Konzept oder Sachverhalt zu *erklären* oder jemanden zu *instruieren* eine Aufgabe ausführen zu können. Die didaktischen Intentionen sind somit durch das Videomaterial vorgegeben, wobei Mischformen innerhalb einer Videoszene oder in Bezug auf die einzelnen Szenen existieren können. In einer Vorlesungsaufzeichnung finden sich beispielsweise Szenen, in denen ein Dozierender Sachverhalte erklärt und andere, in denen ein Procedere erläutert wird. Gleichzeitig kann diese Aufzeichnung auch als Anschauungsobjekt für eine beobachtende Videoanalyse genutzt werden. Für alle drei Intentionen ist es dabei unwesentlich, ob das Videomaterial nur als Ergänzung oder in einem Blended- oder Onlineszenario eingesetzt wird. In welcher Form die Lernenden mit dem Video umgehen und arbeiten, ist dadurch ebensowenig determiniert. Im Verlaufe dieses Kapitels wird auf die unterschiedlichen Lernformen noch genauer eingegangen (siehe Abschnitt 2.4, S. 48).

- **Beobachten:** Durch Aufzeichnung von Verhaltenssequenzen bzw. Verwendung von Beispielsvideos kann das Verhalten der Protagonisten z.B. in Vorträgen, Interviews, Rollenspielen, Diskussionen oder Gesprächssituationen durch die Lernenden beobachtet und daraufhin analysiert, reflektiert und diskutiert werden [Niegemann et al. 2008, S. 266]. Die kollaborative Videoanalyse und -diskussion kann in Präsenz, aber auch in einer virtuellen Lernumgebung stattfinden [Goeze et al. 2014; Pea und Lindgren 2008; Reinhardt et al. 2011]. Mit Verweis auf das „Lernen durch Beobachten“ [Seel und Ifenthaler 2009, S.

37f] können Lernende anhand von Videos auch lernen, das Verhalten einer handelnden Person, gleich einem Modell, nachzuahmen [Niegemann et al. 2008, S. 266]. Das Lernen am Modell wird in vielen *screencasts* intendiert, wenn beispielsweise die Benutzung einer Software vorgeführt wird. Außerdem können auch technische oder physikalische Prozesse als Beobachtungsgegenstand einer Analyse dienen [Börner et al. 2016].

- **Erklären:** Die Vermittlung von Wissen durch das Erklären von Begriffen, Konzepten, Sachverhalten und Zusammenhängen ist ein Haupteinsatzfeld von Lernvideos. Videos dieser Art werden auch als „how-to video“ oder „Erklärvideo“ bezeichnet. Bei den meisten Vorlesungsaufzeichnungen, aber auch bei Dokumentarfilmen handelt es sich um Erklärvideos.
- **Instruieren:** Im Unterschied zu den Erklärvideos enthalten Videoinstruktionen zusätzliche Anweisungen, in welcher Weise ein Vorgang oder eine Handlung auszuführen ist. Beispiel hierfür stellen Software Tutorien in Form von *Screencasts* oder Bewegungsabläufe im Sport dar.

Nicht unerwähnt bleiben sollen an dieser Stelle nachgelagerte Szenarien für den Einsatz von Lernvideos. Auch wenn diese Szenarien nicht vordergründig dem Lernen **anhand** von Video dienen, können sie eine wichtige Funktion innerhalb von Lehr-Lern-Szenarien sowie für die jeweilige Hochschule haben.

- **Lernleistung:** Eine Möglichkeit, die Leistung von Lernenden zu evaluieren, besteht darin, dass die Lernenden selbst ein Video oder Lernvideo produzieren [Alpay und Gulati 2010; Niegemann et al. 2008; Özdener und Güngör 2010; Trahasch 2006; Zahn et al. 2009]. Die Produktion des Mediums sowie die medienkompetente Aufbereitung von Inhalten und Materialien steht dabei im Vordergrund dieser Lernleistung. Der Gegenstand der in diesem Kontext produzierten Videos ist nicht auf die Wissensvermittlung beschränkt.
- **Rahmung:** Mittels audiovisueller Medien lassen sich Einführungen in Online Lernangeboten gestalten, in denen die beteiligten Akteure sich beispielsweise selbst vorstellen, organisatorische Aspekte erläutern oder aber eine problemorientierte Einleitung in Form eines Videos bieten [Niegemann et al. 2008, S. 266]. Durch die persönliche Ansprache (vgl. *Voice Principle* [Mayer 2009]) sowie die Authentizität verbaler und nonverbaler Kommunikation können Lehrende die Lernenden mit Hilfe von Videos aktivieren und motivieren. Das Ziel besteht deshalb oft darin, Lernende für ein Thema zu begeistern oder sie zumindest zur Nutzung eines Lernangebots anzuregen [Schwartz und Hartman 2007].
- **Archivierung:** Videos bieten die Möglichkeit, Wissen zu speichern und somit zu archivieren. Damit stellen sie in der Lehre eine alternative Speicherform zu Präsentationsfolien, Vorlesungsskripten und Büchern dar. An vielen Hochschulen und Bibliotheken wurden zu diesem Zweck digitale Archive für Vorlesungsaufzeichnungen und andere videografische Aufzeichnungen (z.B. Vortragsreden, ePublikationen<sup>2</sup>) geschaffen. Die Ressourcen stehen dann für Recherchen zur Verfügung, ohne dass eine bestimmte (didaktische) Nutzungsform zum Zeitpunkt der Archivierung vorgesehen ist.
- **Marketing:** Lernvideos können dazu beitragen, die Sichtbarkeit, Bekanntheit und Attraktivität einer Bildungs- und Forschungseinrichtung zu erhöhen [Rohs und Streule 2009; Schaffert 2010]. Der Erfolg der Lernvideos wird dabei oftmals durch die Bekanntheit der

<sup>2</sup>Vgl. <http://www.sciencemovies.de/> (abgerufen am 15.10.2016)

darin agierenden Lehrenden und Einrichtungen befördert und wirkt auf ihre Reputation zurück. Beispielhaft dafür sind die *Stanford Open Office Hours*<sup>3</sup>, die Vorlesungsaufzeichnungen bei *iTunes U*, diverse *MOOCs* (z.B. TU9 MOOC der neun großen Technischen Universitäten in Deutschland<sup>4</sup>) sowie videobasierte *Open Educational Ressources*.

Die Bandbreite der Einsatzfelder macht deutlich, dass Lernvideos nicht nur eine didaktische Funktion erfüllen, sondern für Hochschulen, Lehrende und Forschende als Mittel zur Kommunikation mit ihren jeweiligen Zielgruppen dienen. In wie weit dies Auswirkungen auf die Auswahl der im Video verarbeiteten (Lern-)Inhalte hat, sei einmal dahin gestellt. Fest steht jedoch, dass ein Lernvideo im Laufe der Zeit unterschiedlichen Verwendungen zugeführt werden kann. In ähnlicher Weise gilt dies auch für öffentlich zugängliche Videolernumgebungen, die den Anforderungen der unterschiedlichen Zielgruppen gerecht werden müssen, indem sie den Einsatz in unterschiedlichen Kontexten technologisch unterstützen.

### 2.1.3 Potentiale des Medium Video

Im Kontrast zu anderen Online-Lernmedien, die in gleicher Weise orts- und zeitabhängig nutzbar sind, eignen sich Videos insbesondere zur Vermittlung realitätsnaher Situationen und anschaulichen Visualisierung. Insbesondere raumzeitliche Abläufe, dreidimensionale Verhältnisse oder komplexe Bewegungs- oder Interaktionsabläufe lassen sich gut veranschaulichen. Ein wesentlicher Vorzug von Videos resultiert aus dem Vermögen, alle gängigen Symbolsysteme wie zum Beispiel bewegte Bilder aller Realitätsstufen, Standbilder, gesprochene und geschriebene Sprache, Soundeffekte und Musik als kohärentes Medium zu kombinieren [Chambel et al. 2004; Schwan 2005].

Niegemann et al. [2004, S. 148], Zahn [2004], Tiellet et al. [2010], Seel und Ifenthaler [2009], Schwan [2005] und Bates [2015, S. 237-241] heben eine Reihe von Vorteilen der videografischen Informations- und Wissensvermittlung hervor:

**Informationsdichte:** Videos zeichnen sich durch eine hohe Informationsdichte aus. Eine hohe Dichte kann sowohl innerhalb einer Sinnesmodalität (Sehen oder Hören), als auch aus der Kombination visueller und auditiver Modalitäten hervorgehen. Eine visuelle Verdichtung entsteht beispielsweise durch die gleichzeitige Darstellung von Bildern, Text und Bewegtbildern, während das Aufeinandertreffen von Sprache, Musik und Geräusche den Informationsgehalt und die hörbaren Sinneseindrücke verdichten können. Zahn [2004, S. 6]

**Realitätsnähe:** Mit Hilfe von Videos lassen sich realitätsnahe Darstellungen wiedergeben. Realitätsnähe kann sich auf die dargestellten Objekte oder auf die dargestellten Bewegungen, Prozesse und Veränderungen beziehen. Im ersten Fall spricht man von visuell-räumlicher, im zweiten Fall von raumzeitlicher Realitätsnähe. [Zahn 2004, S. 6-7]

**Veranschaulichung dynamischer Veränderungen oder Bewegungen:** Raumzeitliche Abläufe wie z.B. das Wachstum von Pflanzen, exponentielles Wachstum [Bates 2015, S. 238]

<sup>3</sup>Siehe Vorstellung von Philip Zimbardo [http://www.youtube.com/watch?v=x6\\_5lGvIxK0](http://www.youtube.com/watch?v=x6_5lGvIxK0) (abgerufen am 15.02.2015).

<sup>4</sup>Siehe <http://mooc.tu9.de/> (abgerufen am 27.08.2015).



oder die Vorgehensweise beim Knüpfen nautischer Knoten [Riempp 2000] lassen sich mit Hilfe von Video veranschaulichen.

**Anschaulichkeit und Authentizität:** Videos können reale Erfahrungen durch die Darstellung dynamischer Abläufe ersetzen, die entweder nur schwer oder nicht mehr beobachtbar sind oder nicht leicht in Worte zu fassen sind [Tiellet et al. 2010]. Als Beispiel lassen sich etwa aufwendige wissenschaftliche Experimente (z.B. das Milgram-Experiment), Aufnahmen historischer Ereignisse (z.B. die Rede von Martin Luther King, Jr. am 28.08.1963), aber auch nachgestellte Gesprächssituationen zwischen Arzt und Patient oder zwischen Eltern und Lehrer anführen. Die durch Video erreichbare große Anschaulichkeit und Authentizität kann überdies auch textbasierte Lernangebote bereichern [Schwan 2005].

**Zeitbezug:** Die Repräsentation der bildlichen und akustischen Informationen erfolgt sequentiell und in Abhängigkeit der Wiedergabezeit. Die Wiedergabezeit unterscheidet sich dabei von der Erzählzeit im Moment der Aufnahme, d.h. es handelt sich nicht um Echtzeitinformationen. Darüber hinaus lassen sich zeitabhängige Veränderungen durch Animationen oder durch beschleunigte oder verlangsamte Bewegtbilder darstellen [Bates 2015, S. 238].

**Emotionalität:** Zur Erklärung emotionaler Effekte dient das aus der Fernsehforschung bekannte Konzept der Erregung (*arousal*). Je nach Art des Videomaterials kann sich das Erregungsniveau des Rezipienten erhöhen (z.B. Komödien) oder senken (z.B. Natursendungen). Der Grad der Erregung ist durch das Ausgangsniveau der Erregung bestimmt. Je niedriger das Ausgangsniveau, desto größer die Reaktion auf das Informationsangebot. Häufiges Schauen führt zu einer schwächeren emotionalen und affektiven Reaktion. [Seel und Ifenthaler 2009, S. 84f]

**Hinwendungs- und Orientierungsreaktion beim Rezipienten:** „Durch die abrupten optischen Veränderungen der Informationsdarbietung in dynamischen visuellen Medien werden Hinwendungs- und Orientierungsreaktionen beim Rezipienten ausgelöst.“ [Schwan 2005]. Deutlich wird dies beispielsweise, wenn ein Sprecher die Zuschauer persönlich direkt anspricht („Stellen Sie sich einmal vor ...“) oder sie mit rhetorischen Fragen konfrontiert („Wie hätten Sie reagiert?“).

**Interaktionsmöglichkeiten:** Lernende können die Geschwindigkeit der Informationsdarbietung ein Stück weit an ihre kognitiven Fähigkeiten anpassen, indem sie das Video anhalten, langsamer oder schneller abspielen sowie eine Szene wiederholen oder überspringen.

Trotz der genannten Potentiale relativiert Wetzel et al. [1994, S. 184] den Glauben an die Effektivität von Videos als Lernmedium. Die potentiellen Eigenschaften sind viel mehr ein Konglomerat verschiedener Beispiele, von denen sich nur ein Teil, mehr oder weniger gut umgesetzt, in typischen Lernvideos wiederfindet [Wetzel et al. 1994, S. 184]. Die Gestaltung des eigentlichen Videos sowie die Modalität seiner Darbietung in einem Videoplayer oder in einer Lernumgebung hat dabei einen erheblichen Einfluss auf die kognitive Verarbeitung der Informationen und damit auch auf den effektiven Wissenserwerb.

Da die Produktion von Lernvideos nicht Gegenstand dieser Arbeit ist, erfolgt an dieser Stelle lediglich ein Verweis auf die Werke von Thompson und Bowen [2009] und Koumi [2006], die die

Bildsprache des Mediums durch ein „video grammar“ bzw. „visual grammar“ erklären. Darin gehen die Autoren jeweils darauf ein, wie die einzelnen Symbolsysteme z.B. durch Kameraeinstellungen, -position, -winkel, -bewegung und die Komposition der Objekte in einer didaktisch und kognitiv förderlichen Art und Weise aufbereitet werden können.

Neben diesen eher handwerklichen Empfehlungen aus der Filmproduktion gibt es jedoch umfangreiche experimentelle Befunde und Erkenntnisse darüber, wie Menschen audiovisuelle Informationen verarbeiten. Die daraus abgeleiteten kognitiven Lerntheorien betreffen nicht nur Videos als solches, sondern auch die Modalität der sie umgebenden Lernsysteme einschließlich zeitabhängiger, interaktiver Komponenten. Der folgende Abschnitt gibt eine Einführung in medienbezogene Theorien des kognitiven Lernens. Dabei steht die Frage im Mittelpunkt, welche Erkenntnisse aus der Kognitionspsychologie für die Gestaltung und Konzeption von Video-lernumgebungen relevant sind.

### Videos im Kontext kognitiver Lerntheorien

Die kognitiven Prozesse bei der Verarbeitung dynamischer, audiovisueller Darstellungen lassen sich durch die *Theorie der Kognitiven Last* und die *Kognitive Theorie des multimedialen Lernens* erklären und verstehen. Diese Theorien wurden vielfach empirisch geprüft und in einfache Handlungsempfehlungen (Prinzipien) für die Gestaltung multimedialer Lernarrangements überführt. Inwiefern sich daraus Implikationen für die Gestaltung und den Aufbau von Video-lernumgebungen ergeben, soll in diesem Abschnitt geklärt werden.

#### 2.2.1 Theorie der kognitiven Last

Lernmaterialien unterscheiden sich darin, wie weit sie kognitive Kapazitäten im Arbeitsgedächtnis beanspruchen. Bei einer Überanspruchung der Kapazitäten wird der Transfer von Informationen vom Arbeits- ins Langzeitgedächtnis gestört. Der Lernprozess ist damit nicht effektiv. Erkenntnisse der *Cognitive Load Theory* können dabei helfen, unnötige kognitive Belastungen zugunsten freier Ressourcen im Arbeitsgedächtnis bereits im Vorfeld bei der Planung und Konzeption von Lernmedien und Lernumgebungen zu reduzieren.

Man unterscheidet drei Formen der kognitiven Last: (1) die intrinsische Last, die sich aus der Anzahl interaktiver Informationselemente (inkl. deren Beziehungen) in einer Aufgabe ergibt; (2) die *extraneous load* durch Informationen, die dem Lernen nicht zuträglich sind; und (3) *germane load* als die tatsächlich dem Lernen dienliche Information. Lernen ist nur dann effektiv, wenn die gesamte kognitive Last die Verarbeitungskapazität des Arbeitsgedächtnisses nicht übersteigt.

**Intrinsic cognitive load** Die intrinsische kognitive Last ist durch das Lernmaterial bestimmt und kann nicht durch die Art der Wissensvermittlung beeinflusst werden. Der Grad an intrinsischer kognitiver Last ist durch die Anzahl an Elementen bestimmt, die gleichzeitig im Arbeitsgedächtnis verarbeitet werden müssen. Man unterscheidet zudem zwischen voneinander unabhängigen (*low-element interactivity*), seriell zu lernenden Elementen und solchen, die in einem engen Zusammenhang stehen und zum Erlangen eines Verständnisses gleichzeitig das Arbeitsgedächtnis beanspruchen. Dabei steht die Expertise des Lernenden in Relation zum Grad der Elementbeziehungen. Erfahrene Lernende können demnach komplexe Zusammen-



hänge in vorhandene Schemata einordnen, die sie als ein einziges Element im Arbeitsgedächtnis auffassen. Lernenden ohne dieses Vorwissen entsteht eine höhere Belastung, da sie die Elemente einzeln verarbeiten müssen. [Hasler et al. 2007]

**Extraneous cognitive load** *Extraneous cognitive load* ergibt aus der Informationsrepräsentation, welche eine unnötige und ineffektive Belastung des Arbeitsgedächtnisses darstellt. Dies ist insbesondere dann kritisch, wenn die intrinsische Belastung hoch ist. Eine unangemessene didaktische Aufbereitung von Lernmaterialien mit hoher *element-interactivity* kann Lernende leicht überfordern, d.h. mintunter zu schlechten Lernergebnissen führen. Umgekehrt wirken sich inadäquat aufbereitete Lernmaterialien, die nur eine geringe intrinsische Belastung darstellen, nicht negativ auf die Lernleistung aus. [Hasler et al. 2007]

**Germane cognitive load** Die dritte Ursache für eine kognitive Belastung geht aus der Bemühung hervor, aus den Lernmaterialien Schemata zu konstruieren. Im Unterschied zu den zuvor genannten Arten der kognitiven Last, fördert eine hohe *germane load* das Lernen. Die obere Grenze ist dabei die Kapazität des Arbeitsgedächtnisses, die kleiner als die Summe aus *intrinsic*, *extraneous* und *germane load* sein sollte. Angemessene Instruktionsanweisungen reduzieren also die *extraneous load* und erhöhen die *germane load*. [Hasler et al. 2007]

Die Gefahr, die Rezipienten eines Videos kognitiv zu überfordern, entsteht durch den kontinuierlichen Informationsfluss und der begrenzten Kapazität des Arbeitsgedächtnisses des Rezipienten. Eine kognitive Überlastung tritt dann auf, wenn das Tempo der Informationsdarbietung die kognitive Leistungsfähigkeit des Betrachters übersteigt [Merkel et al. 2011].

Eine hohe Informationsdichte ist ein weiterer Aspekt, der zu einer kognitiven Überlastung führen kann. Zum einen ist dies durch die gleichzeitige Präsentation von Artefakten einer Modalität bedingt, wenn z.B. viel Text neben komplexen, dynamischen Animationen zu sehen ist oder Sprache durch Musik oder Geräusche überlagert wird. Zum anderen kann auch aus der Kombination der Repräsentation mehrerer Modalitäten eine kognitive Überforderung entstehen. Dies ist beispielsweise dann der Fall, wenn der Betrachter einen umfangreichen Text lesen soll, während die Sprecherstimme einen anderen Sachverhalt erläutert. Die Vermeidung solcher Effekte wird in den Modalitätsprinzipien auf S. 42 angesprochen.

Auch rasche Bildwechsel, bedingt durch schnelle und kurze Schnitte, erfordern vom Betrachter eine hohe Wahrnehmungsleistung [Niegemann et al. 2004, S. 150], während zu lange Einstellungen bzw. monotone Darbietungen die Aufmerksamkeit schwinden lassen [Wetzel et al. 1994, S.58]. Nachträglich können Steuerungselemente im Videoplayer dem Anwender helfen die beiden letztgenannten Effekte zu reduzieren, um Bilder zeitadäquat zu verarbeiten oder ggf. zu überspringen [Hasler et al. 2007; Mayer 2009]. Aus der Fernsehforschung sind zudem eine Reihe von Techniken bekannt, um die Aufmerksamkeit der Rezipienten aufrecht zu halten: bildliche Komplexität, Bewegung, Schnitte, Kameranäherung, Zoom und Soundeffekte [Wetzel et al. 1994, S.58]. Weidenmann [2002, S. 83f] verweist darüber hinaus auf acht Maßnahmen, die das Risiko des *overloads* vermindern können:

- Verlangsamung der Animation,
- mentale Vorbereitung des Rezipienten ('advanced organizer'),
- Wiederholung,
- Standbildverlängerung,
- strukturierende Schrifteinblendung,
- kongruente und synchrone auditive Kommentierungen,
- Beschränkung der Bewegtbilder auf das erforderliche Mindestmaß,
- *information on demand* („Pull-Funktion“).

### 2.2.2 Kognitive Theorie des Lernens mit Multimedia

Bei der *Kognitiven Theorie des Lernens mit Multimedia* von Richard E. Mayer und Kollegen handelt es sich um eine evidenzbasierte Richtlinie im Bereich des multimedialen Lernens, die auf eine effektivere Gestaltung von dynamischen Visualisierungen, also auch Videos, abzielt. Der Theorie liegen drei Annahmen zugrunde: (1) Menschen verarbeiten in zwei voneinander getrennten Kanälen auditive und visuelle Informationen; (2) jeder dieser beiden Kanäle hat eine begrenzte Verarbeitungskapazität; und (3) Lernen erfordert eine substantielle kognitive Verarbeitung im auditiven und visuellen Kanal [Mayer und Moreno 2003, S. 44]. Darin zu erkennen sind Bezüge zur *Cognitive Load Theory* und *Dual Coding Theory* [Paivio 1990]. Letztere besteht aus der Annahme, dass verbale und bildliche Codes in zwei voneinander unabhängigen Kanälen verarbeitet und Schritt für Schritt im Gedächtnis integriert werden.

Aus der kognitiven Theorie des multimedialen Lernens hat Mayer [2009] zwölf grundlegende Prinzipien für die Gestaltung von multimedialen Lernumgebungen abgeleitet. Aus Sicht der Gestaltung beziehen sich die Prinzipien einerseits auf narrative und visuelle Aspekte der Medienrepräsentation und andererseits auf die Sequenzierung von Lerninhalten. Für die Gestaltung graphischer Benutzerschnittstellen sowie für eine adäquate Usability bieten diese Gestaltungsprinzipien einen theoretischen Bezugsrahmen, der durch randomisierte und kontrollierte Experimente validiert ist. Die experimentellen Belege beruhen jedoch zum großen Teil auf Lernressourcen, die nur einen Teil der dem Video zugeschriebenen Eigenschaften auf sich vereinen. Niegemann et al. [2008] und Zahn [2004] teilen jedoch die Ansicht, dass die evidenzbasierten Richtlinien auf das Lernen mit Videos übertragen werden können. In den nachfolgenden Absätzen werden die Prinzipien im Einzelnen vorgestellt und in Bezug auf das Lernen mit Videos erläutert.

**Kohärenz-Prinzip** Gemäß dem Kohärenzprinzip binden irrelevante Informationen kognitive Ressourcen zur Verarbeitung der relevanten Informationen [Mayer 2009, S. 106]. Aus diesem Grund ist es beispielsweise ratsam, unnötige Grafiken, Bilder, Texte oder akustische Signale zu vermeiden. Falls dies nicht vollständig möglich ist, sollte Lernenden die Unterscheidung von relevanten und irrelevanten Informationen erleichtert werden, indem das Relevante hervorgehoben, zusammengefasst oder den übrigen Informationen vorangestellt wird [Mayer 2009, S. 106f].

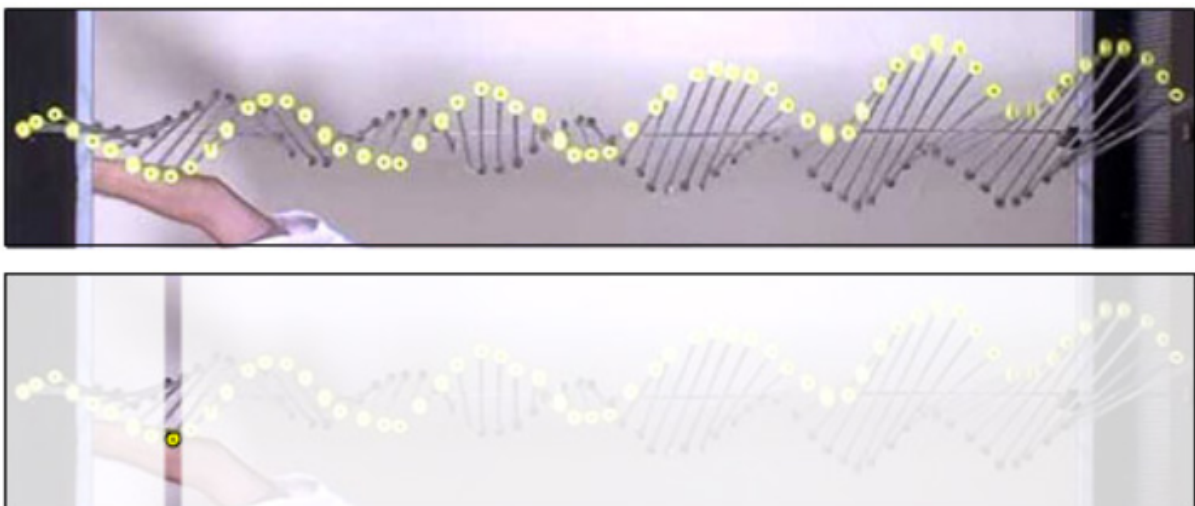
**Prinzip der Signalisierung** Lernende verstehen multimediale Präsentationen besser, wenn darin Hinweise enthalten sind, wie die Informationen zu verarbeiten sind [Mayer und Moreno 2003]. Werden essentielle Informationen hervorgehoben oder betont, fördert dies den Wissenserwerb und mindert gleichzeitig die Verarbeitung irrelevanter Fakten bzw. irrelevanter Aspekte von Grafiken (*extraneous load*). Die Anwendung des Signalisierungsprinzips lenkt die Aufmerksamkeit der Rezipienten auf die Schlüsselemente, die dadurch miteinander in Beziehung gebracht werden. Hervorhebungen bzw. Betonungen unterstützen die mentale Auswahl und Organisation des Lernmaterials. Dies ist umso wichtiger, wenn es nicht möglich ist irrelevante Informationen weg zu lassen, d.h. das Kohärenzprinzip nicht zur Anwendung kommen kann [Mayer 2009, S. 108]. Verbale und visuelle Signale sind jedoch nur dann wirksam, wenn man sparsam von ihnen Gebrauch macht. Abbildung 2.2 zeigt eine Möglichkeit den Fokus der Aufmerksamkeit auf einen Bildausschnitt zu legen. Weitere verbale und visuelle Techniken zur Anwendung des Signalisierungsprinzips in Videos sind Gegenstand der folgenden Aufzählung [Mayer 2009, S. 110/112]:

#### Verbale Signale

- Kurzfassung / Überblick zu Beginn;
- Überschriften mit Bezug zur Kurzfassung;
- sprachliche Betonung durch lautes oder langsames sprechen;
- Schlüsselwörter;
- Signalwörter: z.B. Aufzählungen („erstens, zweitens, ...“);
- Zusammenfassung.

#### Visuelle Signale

- Schaubilder („visual organizer“);
- Auflistungen;
- Pfeile, Rahmen;
- Farbcodes;
- blinkende Schlüsselkomponenten;
- hinweisende Gesten durch den Sprecher;
- zentrale Elemente durch „Ausgrauen“ der Umgebung oder Heranzoomen hervorheben.



**Abbildung 2.2:** Ein Beispiel eines Lernvideos ohne und mit einer Hervorhebung (Quelle: Hatsidimitris und Allen [2010])

**Redundanz-Prinzip** Die redundante, parallele Darstellung von geschriebenem und gesprochenem Text stellt für den Lernenden eine Doppelbelastung dar. Die redundante Information erhöht die kognitive Last durch Irrelevantes und bindet unnötig viele kognitive Ressourcen. Während über den visuellen Kanal Bilder und Texte verarbeitet werden, erfolgt über den auditiven Kanal ein Abgleich des gesprochenen mit dem geschriebenen Wort. Dies ist beispielsweise der Fall, wenn ein Sprecher in einer Vortragsaufzeichnung nur das wiedergibt, was auf den simultan gezeigten Präsentationsfolien zu lesen ist. Dies überfordert den Rezipienten, weshalb die redundanten Informationen zugunsten des gesprochenen Texts zu vermeiden sind. Für dieses Prinzip gelten jedoch eine Reihe von Ausnahmen, falls:

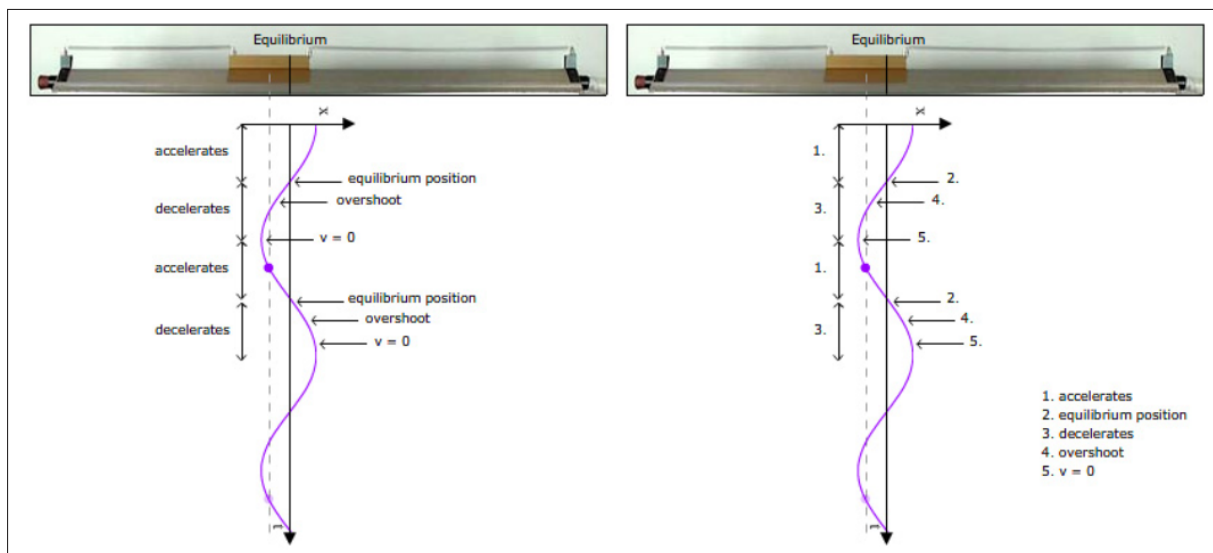
- keine bildlichen Informationen dargeboten werden und der Text kurz ist;
- die Wissenspräsentation sehr langsam erfolgt bzw. durch den Lernenden gesteuert werden kann;
- unbekannte oder fachspezifische Begriffe auftreten;
- der gesprochene Text vor dem angezeigten Text erscheint;
- Nicht-Muttersprachler oder Menschen mit Lernschwierigkeiten zum besseren Verstehen der Lerninhalte Untertitel benötigen. [Mayer 2009, S. 118]

**Prinzip der räumlichen Kontiguität** Es ist lernförderlich, korrespondierende Worte und Bilder nah beieinander zu präsentieren, anstatt sie auf verschiedene Seiten oder Frames zu verteilen. Wenn sich eine enge semantische Beziehung zwischen Informationseinheiten auch durch ihre räumliche Nähe ausdrückt, wird dem Rezipienten die Herstellung dieser Beziehung erleichtert. Dies gilt insbesondere für Relationen zwischen bildlichen und verbalen Informationen, deren kognitive Verarbeitung über den bildlichen Kanal erfolgt. Suchprozesse, die das Arbeitsgedächtnis beanspruchen, lassen sich somit vermeiden. Das Prinzip der räumlichen Nähe ist insbesondere dann anzuwenden, wenn die Darstellung von Illustrationen bzw. Animationen komplex oder nicht selbsterklärend ist. Beispielsweise ist es lernförderlicher, Objekte bzw. Abläufe direkt im Bild zu beschriften, anstatt in einer separaten Legende (siehe Abb. 2.3). Das Prinzip der räumlichen Nähe begründet beispielsweise, warum Videos besser direkt in einer Webseite eingebettet, anstatt verlinkt werden sollten. Lernende mit wenig Vorwissen profitieren besonders von diesem Gestaltungsprinzip.

**Prinzip der zeitlichen Kontiguität** Es ist lernförderlicher, wenn korrespondierende bildliche und narrative Informationen gleichzeitig statt nacheinander präsentiert werden [Mayer 2009, S. 156]. Im Hinblick auf die Entlastung des Arbeitsgedächtnisses erleichtert die Anwendung dieses Prinzips die Bildung mentaler Verbindungen zwischen visuellen und auditiven Repräsentationen.

Das Prinzip kommt jedoch nicht zum Tragen, wenn es sich um kurze Animationen bzw. Videosequenzen handelt, deren Information der Rezipient ohne größere Schwierigkeiten bis zur verbalen Erklärung memorieren kann. Auch wenn es dem Lernenden frei steht, die Reihenfolge und Geschwindigkeit der Darbietung zu bestimmen, ist eine zeitliche Nähe bildlicher und textuell-sprachlicher Informationen nicht unbedingt erforderlich. [Mayer 2009, S. 156ff]

Anwendung findet dieses Prinzips beispielsweise in *screencasts*, in denen ein Sprecher etwa die Verwendung eines Computerprogramms kommentiert und beschreibt. Ein weiteres Bei-



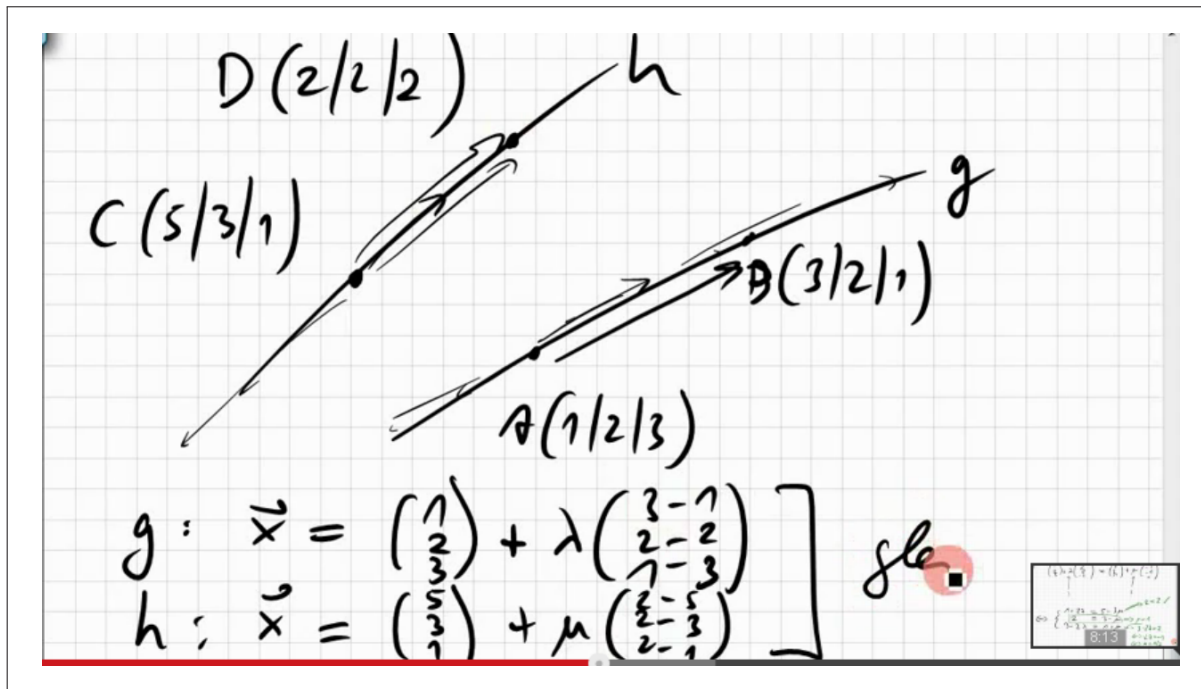
**Abbildung 2.3:** Ein Beispiel eines Lernvideos mit und ohne der räumlichen Nähe von Bild und Text (Quelle: Hatsidimitris und Allen [2010])

spiel für zeitlich synchrone Bild- und Sprachpräsentation sind Präsentationsfolien, die parallel zur Audio- oder Videoaufzeichnung eines Sprechers ablaufen.

**Prinzip der Segmentierung** Präsentationen multimedialer Informationen erzielen eine lernförderlichere Wirkung, wenn sie sequenziell segmentiert sind und ihre Wiedergabegeschwindigkeit und -reihenfolge durch den Lernenden kontrolliert werden können [Mayer 2009, S. 176]. Dieses Prinzip gilt insbesondere für komplexe Lerninhalte, schnell ablaufende Darbietungen und für Lernende mit wenig Vorwissen [Mayer 2009, S. 176ff].

Im Kontext von Lernvideos bedeutet das beispielsweise komplexe Grafiken oder mathematische Formeln, ähnlich wie beim Erstellen eines Tafelbildes, schrittweise zu einem Gesamtbild zu ergänzen (siehe Abb. 2.4). Ein weiteres Beispiel sind kurze, mehrmals abspielbare Videoszenen, die sich als Lernressource besser eignen als eine zusammenhängende Abfolge derselben [Hatsidimitris und Allen 2010]. Innerhalb eines Videoplayers lässt sich dies etwa durch zeitgenaue Sprungmarken eines Inhaltsverzeichnisses erreichen.

**Pre-training Prinzip** Multimediale Präsentationen sind effizienter, wenn den Lernenden Bezeichnungen und Charakteristiken der behandelten Konzepte bereits vorher bekannt gemacht werden [Mayer et al. 2002] (siehe auch [Mayer 2009, S. 191]). Dadurch verbessert sich die kognitive Verarbeitung zur Entwicklung eines kausalen Modells, insbesondere bei komplexen Lehrstoffen, schnell ablaufenden Darbietungen und bei denjenigen Lernenden, die über wenig Vorkenntnisse verfügen. Gegensätzlich wirkt die Anwendung dieses Prinzips für Lernende mit spezifischen Vorkenntnissen. Aus den Studien geht jedoch nur hervor, dass die Lernenden durch die vorherige Einführung von Begriffen und Konzepten mehr lernen und behalten, als wenn sie keine oder erst eine nachträgliche Einführung erhalten [Mayer et al. 2002].



**Abbildung 2.4:** Beispiel eines Lernvideos, bei dem Formeln per Stifteingabe auf einem *tablet computer* zusammen mit dem Tonsignal aufgezeichnet wurden (Quelle: Screenshot aus Jörn Loviskach: „01B.3 Schnittpunkt zweier Geraden in 3D, Geradengleichung“, 4:01, <http://youtu.be/2wC6AXoAQfc> (abgerufen am 15.02.2015)).

**Modalitätsprinzip** Audiovisuelle Darstellungen bildlicher und narrativer Informationen fördern den Wissenserwerb mehr als die alleinige visuelle Repräsentation der gleichen Informationen [Mayer 2009, S. 200]. Die Kombination aus auditiven und visuellen Informationen empfiehlt sich insbesondere für komplexe Lehrstoffe, schnelle Bildabfolgen und falls die Lernenden mit den in der Darbietung vorkommenden Begriffen bereits vertraut sind. In bestimmten Ausnahmen ist die Darstellung von Text gegenüber der Kombination von Bild (Animation) und Narration vorzuziehen. Beispielsweise dann, wenn viele Fachbegriffe oder Symbole verwendet werden, Faktenwissen auswendig zu lernen ist oder die Lernenden keine Muttersprachler sind.

**Multimediaprinzip** Integrative Medienrepräsentationen aus narrativen bzw. geschriebenen Wort und Bildern sind lernförderlicher als auditive Medienpräsentationen [Mayer 2009, S. 223]. Bleibt der visuelle Kanal ungenutzt, besteht die Gefahr kognitive Ressourcen zu verschenken, sodass diese sogar kontraproduktiv eingesetzt werden könnten. Laut diesem Prinzip sind Videos im Vergleich zu Tonaufnahmen besser geeignet, Lerninhalte zu vermitteln. Voraussetzung für die Wirksamkeit multimedialer Repräsentationen ist qualitativ hochwertiges und didaktisch gut aufbereitetes Bildmaterial. Effektiv ist dieses Gestaltungsprinzip bei Zielgruppen mit wenig Vorwissen.

**Personifizierungsprinzip** Persönliche Ansprachen sind wirksamer als eine formale Sprechweise [Mayer 2009, S. 242]. Ausgehend von der Annahme, Lernen mit Multimedia entspreche einer sozialen Interaktion zwischen Lehrenden und Lernenden, reagieren Lernende auf eine persönliche Ansprache. Beispielsweise sind in einer Ansprache die Personalpronomen der ers-



ten und zweiten Person gegenüber den der dritten Person zu bevorzugen. Ein weiteres Stilmittel besteht in der Adressierung von Aussagen der Art: „Stell dir vor, du bist . . .“ oder „Wie Sie sehen, . . .“. Des Weiteren geht aus diesem Prinzip hervor, dass Designeffekte bei Lernenden mit geringem Vorwissen eine größere Wirkung zeigen, als bei Lernenden mit hohem Vorwissen. Letztere sind im Stande Unzulänglichkeiten dieser Art auszugleichen. [Mayer 2009, S. 243] Da dieses Prinzip durch den Audiokanal des Videos determiniert ist, besteht innerhalb einer Lernumgebung kaum eine Möglichkeit den beschriebenen Effekt nachträglich herbeizuführen.

**Prinzip der Stimmauswahl** Rezipienten reagieren auf soziale Kontextinformationen wie z.B. Geschlecht, Herkunft und Stimmungslage, die aus der Stimme hervorgehen, empfindlich. Da Methoden der Sprachsynthese diese Nuancen gegenwärtig noch nicht hinreichend imitieren, gilt es, synthetische Stimmen zu vermeiden. Die menschliche Stimme ist in auditiven Medien lernförderlicher als synthetische Stimmen [Mayer 2009, S. 257].

**Bildprinzip** Das Bildprinzip besagt, dass es nicht zwingend den Lerneffekt fördert, wenn der Instruktor auf dem Bildschirm zu sehen ist [Mayer 2009, S. 258ff]. Das Prinzip gilt für reale und animierte Sprecher gleichermaßen. Verbreitet ist eine solche Art der Darbietung beispielsweise in *WBTs*, wo der Sprecher neben den Präsentationsfolien zu sehen ist. Bei der gleichzeitigen Darbietung von zwei visuellen Medienrepräsentationen kann der so genannte *split attention effect* auftreten, d.h. der Lernende folgt dem Lehrenden, anstatt die inhaltlich relevanteren Grafiken oder Texte in der parallelen Darstellung zu rezipieren [Mayer 2009, S. 258ff]. In Anbetracht des Signalisierungsprinzips kann der Lehrende durch seine bildliche Repräsentation jedoch gewisse Akzente setzen und durch seine Erscheinung motivieren.

In diesem Abschnitt stand die Frage im Mittelpunkt, welche Faktoren aus kognitionspsychologischer Sicht die Lernförderlichkeit von Videos beeinflussen. Aus den Theorien und den daraus abgeleiteten Gestaltungsprinzipien des multimedialen Lernens ergeben sich eine Reihe von Gestaltungsempfehlungen für Videos und Videolernumgebungen.

Seel und Ifenthaler [2009, S. 126] weisen jedoch darauf hin, dass die Prinzipien auf Untersuchungen von sehr kurzen Lernphasen von weniger als 20 Minuten beruhen, was ihre „Generalisierbarkeit über Lernsituationen erheblich einschränkt“ [Seel und Ifenthaler 2009, S. 126]. Mayer [2009] berücksichtigt zudem weder rezeptive, selbstgesteuerte, noch kollaborative Lernformen und die dafür relevanten motivationalen Aspekte. Auch die Frage danach, wie Lernende ihr Handeln innerhalb eines umfangreichen audiovisuellen Wissensbestands organisieren und selbst steuern, bleibt offen. „Diese Designprinzipien“, schreibt Zahn [2004], „wurden jedoch ausschließlich an linearen audiovisuellen Animationen untersucht und sind spezifisch für Aufgaben, bei denen ein vertieftes Verstehen von Kausalzusammenhängen und der Transfer des Gelernten auf neue Situationen gewährleistet sein sollen“ [Zahn 2004]. Gestaltungsvariationen von Lernprogrammen, in denen das (lineare oder nicht-lineare) Video eine tragende Rolle in der Wissensvermittlung spielt, führt Mayer [2009] nicht weiter aus.

Ungeachtet dieser Kritik geben die kognitiven Theorien Aufschluss über die lernförderliche Gestaltung von Lernvideos. Grundsätzlich können diese Prinzipien bereits bei der Planung und Produktion eines Videos berücksichtigt werden. Bestimmte Gestaltungselemente lassen sich

**Tabelle 2.1:** Aus den 12 Prinzipien des kognitiven Lernens mit Multimedia abgeleitete Anforderungen an Videolernumgebungen.

Prinzip	Anforderung
Segmentierungsprinzip	Inhalte segmentieren bzw. zeitgenauen Zugriff auf Informationen gewährleisten – beispielsweise durch Kapitelmarken oder <i>tags</i> .
Signalisierungsprinzip	Visuelle Hervorhebungen durch Pfeile, Rahmen, Heranzoomen oder Ausgrauen.
Redundanzprinzip	Option das Videobild auszublenden bzw. den Ton abzuschalten (falls Narrativ gleich dem geschriebenen Wort).
Räumliche Kontiguität	Beschriftungen sollten nahe am korrespondierenden Objekt erfolgen. Medien sollten nach Möglichkeit eingebettet, statt verlinkt werden.
Zeitliche Kontiguität	Synchrone Darstellungen von Präsentationsfolien zum Videobild oder zum Ton eines Sprechers. Zeitlich genaue Verankerung von Annotationen, wie z.B. Kommentaren oder Untertitel, im Video.
Räumliche / zeitliche Kontiguität	Einblenden von Überschriften.

jedoch auch nachträglich innerhalb der Lernumgebung implementieren. Dies kann erforderlich sein, wenn durch ein Video eine hinderliche kognitive Belastung entsteht. Beispielsweise lassen sich in einer Lernumgebung Vorkehrungen treffen, um einen Redundanzeffekt durch Ausblendung von Text oder Stummschaltung des Tons zu vermeiden. Aber auch die Wiederverwendung eines Videos in einem anderen didaktischen Szenario oder für eine andere Zielgruppe kann nachträgliche Anpassung erfordern. In Tab. 2.1 sind daher Anforderungen für die Gestaltung von videobasierten Lernumgebungen zusammengefasst, die sich aus den zwölf Prinzipien des kognitiven Lernens mit Multimedia ableiten lassen.

Neben „gestalterischen Eingriffen zur Erleichterung des Lernens [...] [stehen damit] auch dem Lernenden selbst Mittel zur Verfügung eine eventuell hohe kognitive Belastung beim Lernen mit Multimedia-Videos zu kompensieren“ [Zahn 2004]. Diese Mittel bieten insbesondere interaktive Videos. Der folgende Abschnitt thematisiert deshalb das Konzept der Interaktivität in Bezug auf den Wissenserwerb mit audiovisuellen Medien.

## Interaktivität audiovisueller Lernmedien

Der Begriff der Interaktivität bezieht sich hier auf die Schnittstelle zwischen Anwender und Computer, nicht jedoch auf die computervermittelte Interaktion zwischen Menschen, auf am Ende dieses Kapitels behandelt wird (siehe Teilabschnitt 2.4.3, S. 52).

Da moderne Mensch-Computer-Schnittstellen per Definition interaktiv sind, mag das Konzept der Interaktivität computerbasierter Medien zunächst wie eine Tautologie anmuten [Manovich 2001, S. 55]. Dennoch lassen sich verschiedene Grade an Interaktivität feststellen, die es dem Nutzer erlauben die Informationsdarbietung zu regulieren [Zahn 2004] oder durch Eingriffe zu manipulieren [Schwan 2005]. Eine ordinale Skalierung der Interaktivität auf Grundlage der Häufigkeit, Qualität oder dem medialen Charakter der Interaktion schafft jedoch keine Vergleich-



barkeit zwischen Lernumgebungen. In den ersten beiden Teilabschnitten werden daher die Vorschläge von Schulmeister [2005] und Riempp [2000] aufgegriffen, die die Interaktivität elektronischer Lernmedien in zwei qualitativen Kategorierahmen beschreiben. Neben diesen eher theoretischen Fragestellungen, sollen im letzten Teilabschnitt Untersuchungen zum Nutzen und zur Lernförderlichkeit interaktiver Videos dargestellt werden.

**Taxonomie der Interaktivität** Schulmeisters Taxonomie umfasst sechs Stufen und reicht von der Informationsdarbietung bis zur qualifizierten Rückmeldung des Lernprogramms an den Lernenden. In der folgenden Aufstellung werden die sechs Interaktionsstufen anhand von Beispielen interaktiver Videos erläutert.

1. **Objekte betrachten und rezipieren:** Vorgefertigte Videos werden abgespielt, wobei der Betrachter keinen Einfluss auf die Darstellungsform, Abspielposition und den Inhalt ausüben kann. Die Medienelemente dienen der Illustration oder Information. Eine vergleichende Untersuchung dieser nicht-interaktiven Form eines Videos mit der dritten Interaktionsstufe findet sich bei Schwan und Riempp [2004].
2. **Multiple Darstellungen betrachten und rezipieren:** Die multimedialen Komponenten sind ebenfalls vorgefertigt, durch Benutzerinteraktionen können jedoch Teile davon ausgetauscht werden. Beispielsweise kann der Nutzer Videos aus einer Kollektion auswählen und beliebig oft abspielen. Die Medienelemente dienen der Illustration oder Information.
3. **Variation der Repräsentationsform:** Benutzer haben einen aktiven Einfluss auf die Informationsdarbietung. Es kann jedoch nur die Repräsentation, nicht das Ausgangsmaterial manipuliert werden. Beispielsweise können Benutzer ein Video skalieren, drehen oder zoomen sowie anhalten, abspielen und verlangsamen/beschleunigen. Ein weiteres Beispiel ist die nicht-lineare Navigation in Videos, in denen man mit Hilfe zeitbasierter Hyperlinks Segmente im selben bzw. in einem anderen Video aufruft. Diese Form der Interaktivität findet sich mittlerweile in vielen Videolernumgebungen. Exemplarisch sei hier nur auf die Studie von Schwan und Riempp [2004] verwiesen.
4. **Inhalt der Komponente beeinflussen:** Anwender erzeugen neue Objekte, indem sie Daten oder Parameter eingeben. Auf diese Weise können Nutzer beispielsweise Parameter in mathematischen Formeln anpassen und erhalten im Ergebnis eine grafische Ausgabe der Funktion. Videos lassen sich nicht ohne Weiteres durch Änderung von Datensätzen oder Parametern generieren. Ein Beispiel dafür stellt jedoch die Applikation DIVER<sup>TM</sup>/WebDIVER<sup>TM</sup> dar [Pea 2004], bei der Nutzer spatiale und temporale Ausschnitte von 360°-Aufnahmen wählen und annotieren [Pea 2006]. Prinzipiell lassen sich Videos auch durch Eingabe von Texten bzw. Verweisen auf Ressourcen (URI) mit Hilfe von Verfahren der semantischen Textanalyse, der Sprachsynthese und durch eine Bildersuche generieren. Auch wenn diese Verfahren gegenwärtig noch nicht in Lernszenarien Verwendung finden, bestätigen Prototypen die prinzipielle Machbarkeit [Laker 2013; Vaucelle et al. 2003].
5. **Objekte und Prozesse generieren:** Das Lernprogramm wird zum Werkzeug, mit dem die Anwender ihre Ideen in konkrete Modelle, Objekte oder Prozesse überführen können. Diese konstruktive und eigenaktive Form des Lernens kann in Bezug auf die Produktion

(z.B. *Kaltura* [Kaltura-Corporation 2016]) oder Annotation (z.B. mit dem *Mozilla Popcorn Maker* [Mozilla-Foundation 2012]) von Videos realisiert werden.

6. **Konstruktion mit situationsabhängigen Rückmeldungen:** Das Programm versteht Benutzerinteraktionen sowie die daraus entstandenen Objekte und reagiert darauf mit bedeutungsvollen Rückmeldungen. Adaptive Lernwegsverzweigungen können beispielsweise mit *XIMPLE* [Bhikharie et al. 2016] realisiert werden.

**Mikro- und Makrointeraktivität** Alternativ zu Schulmeisters Categoriesystem definieren Riempff [2000], Schwan [2005], Zahn [2004] und Merkt et al. [2011] die Interaktivität von Lernvideos anhand von zwei Bezugsrahmen. Sie unterscheiden mit der Mikro- und Makrointeraktivität Eingriffsmöglichkeiten zum einen innerhalb eines Videos und zum anderen bezogen auf ein oder mehrere Videos als Ganzes.

„Mikrointeraktivität bezeichnet die Veränderbarkeit der Informationsdarbietung innerhalb des Videos“ [Schwan 2005]. Diese umfasst die Steuerungsfunktionen (z.B. Play, Pause, Vorlauf, Rücklauf, Zeitlupe) sowie zeitliche Verweise innerhalb des Videos oder solche, die zusätzliche Informationen einblenden. Riempff [2000, S. 62] unterscheidet darüber hinaus die *temporale Mikrointeraktivität*, die Eingriffe auf die zeitliche Wiedergabe umfasst, von der *spatialen Mikrointeraktivität*, die die Manipulation der räumlichen Darstellung des Videos betrifft.

Interaktivität auf Makroebene baut auf den Funktionen der Mikroebene auf. „Makrointeraktivität umfasst alle Steuerungsmöglichkeiten, die Videosequenzen als Ganzes betreffen, ohne die Videosequenzen selbst in ihren Darbietungseigenschaften zu verändern.“ Riempff [2000, S. 61]. Dies beinhaltet den wahlfreien Zugriff und eine beliebig häufige Nutzung von Videos innerhalb der medialen Gesamtstruktur. Folgende weitere Eingriffsmöglichkeiten repräsentieren die Makrointeraktivität:

- Suche und Auswahl von Videos,
- Auswahl von Videos oder Lernpfaden entlang von Videos [Zahn 2004],
- Aufruf eines Videos durch einen Hyperlink [Riempff 2000, S. 61],
- Verzweigung zwischen Videoszenen [Riempff 2000, S. 61],
- Änderung der Reihenfolge einer Sequenz von Videos [Riempff 2000, S. 61].

**Lernförderlichkeit und Nutzen interaktiver Videos** Interaktionen zwischen Anwendern und Video werden durch das umgebende Videosystem ermöglicht und müssen dazu in dem System implementiert sein. Da die Implementierung in der Regel mit Kosten und Aufwand verbunden ist, stellt sich u.a. die Frage nach dem didaktischen Nutzen. Darüber hinaus gilt es auch zu bedenken, dass ein Überangebot von Funktionen eine kognitive Belastung darstellen kann oder vom eigentlichen Lerngegenstand ablenkt. Wie förderlich das Lernen mit interaktiven Videos im Vergleich zu nicht oder weniger interaktiven Video ist, wurde auf dem Mikro-Level bereits in mehreren Studien untersucht. Die Ergebnisse sprechen überwiegend für die Verwendung von interaktiven Videoplayern, wobei sehr unterschiedliche Stufen an Interaktivität betrachtet wurden. Obwohl die nachfolgende Aufstellung keinen Anspruch auf Vollständigkeit erhebt, konstatieren Merkt et al. [2011] sowie Zahn [2004] einen Forschungsbedarf.

**Wetzel et al. [1994]** betrachteten in mehreren Studien den Nutzen von interaktiven Videos in der Lehre. Die Autoren belegen die lernförderliche Wirkung interaktiver Videos beim Erlernen von Fakten und Konzepten sowie von Fähigkeiten und Bewegungsabläufen (vgl. [Zahn 2004, S. 26]).

**Schwan und Riempp [2004]** untersuchten anhand von 14 - 35 Sekunden langen Clips das Anfertigen von vier Schifferknoten in einem Video ohne Steuerungselemente und in einem interaktiven Video mit Steuerungsinstrumenten (inkl. Anpassung der Wiedergabegeschwindigkeit und -richtung). Zum Erlernen von Techniken zum Knüpfen von Knoten benötigten die Probanden mit Hilfe des interaktiven Videos weniger als zwei Drittel der Zeit als die Probanden mit dem statischen Video.

**Zhang et al. [2006]** konnten in einer vergleichenden Untersuchung zeigen, dass interaktive Videos mit nicht-linearen Interaktionsmöglichkeiten zu besseren Lernergebnissen führen und die Zufriedenheit der Lernenden erhöhen. Das interaktive Video bestand aus einem Sprechervideo, welches mit Präsentationsfolien und einem Transkript synchronisiert war. Lernende können durch die Folien navigieren oder auf ein Inhaltsverzeichnis zugreifen.

**Zahn et al. [2010]** untersuchten die Lösung von Gestaltungsaufgaben mit Hilfe von *WebDIVER* und einen herkömmlichen Videoplayer im Zusammenspiel mit einem Texteditor. *WebDIVER* gestattete den Lernenden, den Film zeitlich und räumlich zu segmentieren, zu editieren und zu annotieren. Im Ergebnis förderte die interaktive Videoanwendung (1) das Verständnis des Lerngegenstands und den Erwerb kognitiver Fähigkeiten, (2) die Qualität der gestalteten Produkte und (3) die Effizienz der Zusammenarbeit in Zweiergruppen.

**Merkt et al. [2011]** verglichen das Lernen mit einem illustrierten *eBook* mit einem gewöhnlichen und einem erweiterten Videoplayer zunächst im Labor und dann im Feld. Der gewöhnliche Player ermöglichte das Abspielen und Anhalten des Videos sowie das Vor- und Zurückspulen. Der interaktive Videoplayer umfasste folgende Funktionen: Start/Stopp, Zeitleiste mit Kapitelmarkierungen, Inhaltsverzeichnis, Schlagwortverzeichnis. Die Studienteilnehmer konnten die Medien 90 Minuten lang nutzen. Anschließend absolvierten sie einen *Multiple-Choice-Test* und schrieben einen Aufsatz zum Thema. In beiden Studien führte die Benutzung des gewöhnlichen Videoplayers zu den besten Lernergebnissen beim Schreiben des Aufsatzes. Die Ergebnisse des *Multiple-Choice-Tests* zeigten keine Effekte hinsichtlich der Medienwahl.

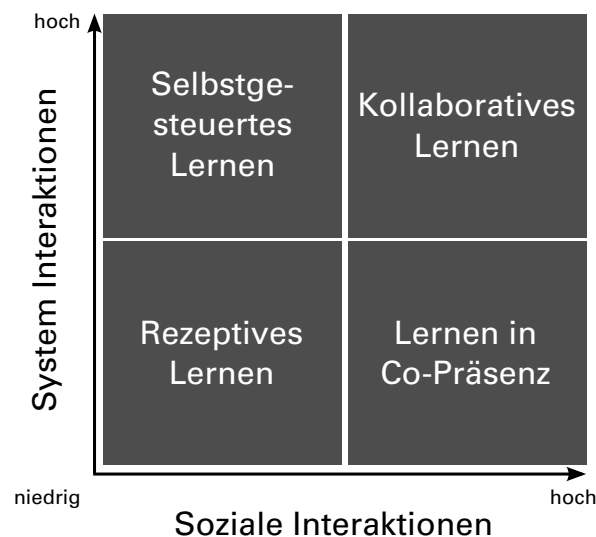
Der zurückliegende Abschnitt behandelte die Interaktivität von Lernvideos. Dabei wurden Belege für die Lernförderlichkeit zitiert und zwei geeignete Klassifikationssysteme zur Beschreibung interaktiver Videos vorgestellt. Obwohl beide Klassifikationen Parallelen aufweisen, lassen sie sich nur schwer vereinen. Die alleinige Beschränkung auf Interaktionen zwischen dem Anwender und dem Videosystem wird den gegenwärtigen Nutzungsgewohnheiten im videobasierten Web 2.0 nur teilweise gerecht. Soziale Interaktionen anhand eines Videos, wie sie beispielsweise Pea und Lindgren [2008] als Ergebnis kollaborativer Aktivitäten aufzeigen, waren von der Betrachtung ausgeschlossen. In Bezug auf das Web 2.0 geht zumindest Schulmeister auf den Übergang vom Rezipieren zum Produzieren ein und adressiert damit den typischen Rollenwechsel vom Konsumenten zum Produzenten. Nicht der Produzent, sondern der Betrachter synthetisiert eine lineare Informationsfolge, indem er Videos auswählt, Teile davon betrachtet

und per Link zum nächsten Video gelangt [Schwan 2005; Vardi 1999]. Dieser Übergang lässt sich auch anhand von Lernformen erkennen, deren didaktisches Design je nach eingesetzter Technologie auf Lehrende, einzelne Lernende oder Lerngruppen ausgerichtet ist. Im nächsten Abschnitt werden deshalb unterschiedliche Lernformen hinsichtlich ihrer didaktischen und technischen Merkmale in Bezug auf das Lernen mit Videos betrachtet.

### Lernformen

Als Lernform bezeichnet man eine „durch bestimmte Merkmale gekennzeichnete Vorgehensweise des Lernens“ [Tenorth und Tippelt 2012, S. 479]. Im Hinblick auf das computergestützte Lernen mit Videos gibt es mit dem rezeptiven, dem selbstgesteuerten und dem kollaborativen Lernen sowie dem Lernen in Co-Präsenz vier wesentliche Formen.

Bei den ersten drei Formen handelt es sich um ortsunabhängige, d.h. E-Learning-gestützte Lernformen. Co-Präsenz bedeutet hingegen, dass mehrere Personen zur gleichen Zeit am gleichen Ort ein Lernmedium nutzen (z.B. Krueger [2010]; Zahn et al. [2010]). Typischerweise finden sich diese Lernform in Präsenzveranstaltungen wie etwa Vorlesungen oder Seminaren. Die Steuerung des Videos obliegt dabei einer einzigen Person (i.d.R. dem Dozierenden). Die Anwesenden rezipieren das Video oder einen Teil davon, bevor sie bestimmte Aspekte oder Fragestellungen miteinander diskutieren und Ergebnisse festhalten. Da diese Lernform keinen direkten Bezug zum E-Learning aufweist, soll sie im Rahmen dieser Arbeit keine weitere Beachtung finden.



**Abbildung 2.5:** Schematische Übersicht der vier grundlegenden videobezogenen Lernformen in Anbetracht der Interaktion mit anderen Lernenden und dem Videosystem

Die vier genannten Gruppen von Lernformen unterscheiden sich, wie in der schematischen Abb. 2.5 zu sehen ist, hinsichtlich des Ausmaß an Interaktionen mit dem System und den beteiligten Personen. Warum dies so ist und welche Anforderungen sich daraus für Videolernumgebungen ergeben, soll in diesem Abschnitt erläutert werden.

### 2.4.1 Rezeptives Lernen

Beim rezeptiven Lernen mit Videos nehmen die Lernenden eine mehr oder weniger passive Rolle ein. Die Rezeption von Lerninhalten ist die Konsequenz ihrer Darbietung im Video. Der dahinter stehende didaktische Ansatz gründet auf der Bereitstellung und Distribution von Informationen [Kay 2012]. Diese Lernform ist durch die Funktionalität der verwendeten technischen Systeme bedingt, indem kaum eine Unterstützung für den Lernprozess durch Lehrende angeboten bzw. durch Lernende in Anspruch genommen wird. Das seit den 1960er Jahren bestehende Bildungsfernsehen (vgl. [Bates 1984; Wetzel et al. 1994]) ist dabei ein typischer Vertreter für ein solches „broadcasting“-System. Aktuelle Beispiele stellen Archive für Vorlesungsaufzeichnungen (z.B. *NPTEL*<sup>5</sup>), Videos in Ablageordnern eines LMS oder CMS sowie Podcasts (z.B. *iTunes U*<sup>6</sup>) [Kay 2012] dar.

**Vorzüge** Im Hinblick auf organisatorische Aspekte des Lehrens sind rezeptive Formen der videobasierten Wissensvermittlung flexibel und einfach umzusetzen, indem die Videos beispielsweise als ergänzende Lernmaterialien (vgl. Teilabschnitt 2.1.2, S. 32) angeboten werden. Ein einmal bereitgestelltes Video kann ohne eine tutorielle Betreuung von einer beliebigen Anzahl an Lernenden orts- und zeitunabhängig genutzt werden. Das Video selbst lässt sich überdies gut mit anderen Lernformen kombinieren und wiederverwenden. Aus Sicht der Lernenden sind Videos *on demand* verfügbar. Zudem ist ein wahlfreier Zugriff auf die zeitlich strukturierten Informationen möglich. Einzelne Szenen lassen sich beliebig oft wiedergeben.

**Limitationen** Die kategorische Form des rezeptiven Lernens in einer videobasierten Lernumgebung bietet Lernenden jedoch keinen Rückkanal für Feedback oder Fragen sowie keine Möglichkeit, den eigenen Lernfortschritt zu überprüfen. Die Rezipienten nehmen somit keinen Einfluss auf die Wissensvermittlung und können innerhalb der Lernumgebung nicht aktiv mitarbeiten oder mit anderen zusammenarbeiten (vgl. [Krueger 2010, S. 18]). Ergebnisse der medienpsychologischen Forschung zeigen, dass die alleinige Betrachtung audiovisueller Medien für die genaue Erschließung der enthaltenen Informationen nicht ausreicht [Spiro et al. 2007]. Außerhalb des E-Learning-Systems, welches das Video in einem Player ausliefert, ist eine aktive Auseinandersetzung etwa durch das Mitschreiben oder Recherche natürlich nicht ausgeschlossen. Die Lernumgebung unterstützt und fördert den Lernenden jedoch nicht dabei.

**Unterstützung durch Videolernumgebung** Um den Zugang zu den zeitbasierten Informationen noch weiter zu verbessern, ermöglichen typische Systeme dieser Art die Anpassung der Wiedergabegeschwindigkeit sowie den gezielten Zugriff auf einzelne Abschnitte durch ein Inhaltsverzeichnis. Auch die synchrone Darstellung weiterer Medien, wie z.B. Präsentationsfolien, ist verbreitet. Neben diesen Interaktionsmöglichkeiten auf Mikroebene, repräsentieren Suchfunktionen und kategoriebasierte Übersichtsseiten die Makrointeraktivität. Nicht vorgesehen sind Funktionen zur Anreicherung von Informationen durch die Lernenden sowie Mittel zur Kommunikation untereinander.

<sup>5</sup>Siehe *National Programme on Technology Enhanced Learning* (Indien): [nptel.ac.in/courses.php](http://nptel.ac.in/courses.php) (abgerufen am 15.02.2015).

<sup>6</sup>Vgl. <http://www.apple.com/de/education/itunes-u/> (abgerufen am 15.02.2015).

### 2.4.2 Selbstgesteuertes Lernen

Für das selbstgesteuerte Lernen werden unterschiedliche Begriffe verwendet, die jedoch oft auf ähnliche Konzepte hinauslaufen. In der englischsprachigen Literatur ist beispielsweise von *self-directed*, *self-regulated*, *explorative* oder auch *Life Long Learning* die Rede. Im Deutschen spricht man auch vom selbst-regulierten, autodidaktischen, entdeckenden oder eigenständigen Lernen. Arnold et al. [2000, S. 17] (zitiert nach Krueger [2010, S. 42]) definiert das selbstgesteuerte Lernen als einen aktiven Aneignungsprozess, „bei dem das Individuum über sein Lernen entscheidet, indem es die Möglichkeit hat,

- die eigenen Lernbedürfnisse bzw. seinen Lernbedarf, seine Interessen und Vorstellungen zu bestimmen und zu strukturieren,
- die notwendigen menschlichen und materiellen Ressourcen (inklusive professionelle Lernangebote oder Lernhilfen) hinzuzuziehen,
- seine Lernziele, seine inhaltlichen Schwerpunkte, Lernwege, -tempo und -ort weitestgehend selbst festzulegen und zu organisieren,
- geeignete Methoden auszuwählen und einzusetzen und
- den Lernprozess auf seinen Erfolg sowie die Lernergebnisse auf ihren Transfergehalt hin zu bewerten.“

**Vorzüge** Selbstgesteuertes Lernen kann sowohl in formellen, als auch in informellen Lehr-Lern-Szenarien stattfinden. Im Unterschied zum rezeptiven Lernen üben die Lernenden einen Einfluss auf die Wissensvermittlung aus, indem sie aktiv durch die Bearbeitung von Aufgaben und Tests [Kerres 2012, S. 7], allein oder in einer Gruppe, mitwirken. Als Beispiel dafür kann man zahlreiche *MOOCs* (z.B. *coursera*, *edX* oder *Udacity*<sup>7</sup>) heranziehen, in denen solche Tests in die zeitliche Struktur einzelner Themenblöcke eingeflochten sind.

**Limitationen** Am Beispiel der *MOOCs* wird aber auch deutlich, dass die Selbstregulierung nicht allein durch Multiple-Choice-Tests erreicht werden kann. Schulmeister [2014] weist zu recht darauf hin, dass die stark segmentierten und durch Testaufgaben unterteilten *MOOC*-Videos behavioristischen Lernprogrammen im Sinne der „programmed instructions“ ähneln. Dieser monotone Wechsel von Wissensvermittlung und Prüfung kann demotivierend wirken, da Kontakte zu Lernenden und Lehrenden, wie auch Diskussionen bestenfalls in anderen virtuellen Räumen stattfinden.

Um also von den Möglichkeiten des selbstgesteuerten Lernens Gebrauch machen zu können, muss der Lernende ausreichend Motivation und auch das nötige Vorwissen besitzen. Außerdem bedarf es bestimmter Fähigkeiten zur Selbstorganisation, denn eine „fehlende Fremdsteuerung“, so Kerres [2012, S. 23 ff], „führt nicht automatisch zur Selbststeuerung!“

In Tab. 2.2 sind Kategorien von Fähigkeiten zusammengefasst, die mit dem Lernerfolg korrelieren. Finke [2005] schätzt die Anforderungen an einen Lernenden hinsichtlich dieser Lernform als sehr hoch ein. Er weist auf eine mögliche Überforderung ungeübter Lernender hin, wodurch

---

<sup>7</sup>Vgl. <https://www.coursera.org/>, <https://www.edx.org/> und <https://www.udacity.com/> (jeweils abgerufen am 15.02.2015).

sich die Lernwirksamkeit mindert. Aus diesem Grund sollte diese Lernform eingeübt und durch direkte Förderansätze angeregt und aufgebaut werden [Kerres 2012, S. 23 ff].

**Tabelle 2.2:** Konzepte des selbstgesteuerten Lernens (in Anlehnung an Straka et al., 1996, S. 154, zitiert nach Krueger [2010]), ergänzt um mögliche Funktionalitäten zur Unterstützung derselben in einer Videolernumgebung am Beispiel von *YouTube*.

Kategorie	Konzept	Beschreibung	Unterstützung in YouTube
Aneignen	Elaborieren	an bestehendes Wissen anknüpfen	Suche, Ähnliche Video
	Strukturieren	Erstellen von Gliederungen, Skizzen, Übersichten, Zusammenfassungen	Playlists, Lesezeichen
	Wiederholen	Lerninhalte wiederholen	Übersicht der betrachteten Videos/Sequenzen
Kognitive Kontrolle	Konzentration	Aufmerksamkeit und Intensität der Informationsverarbeitung steuern	Hervorhebungen im Video
Metakognitive Kontrolle	Reflektieren	den vollzogenen Lernweg reflektieren	Historie der betrachteten Videos
	Regulieren	einen anderen Lernweg verwenden	auf ähnliche, grundlegende oder weiterführende Videos verweisen
	Überwachen	auf den aktuellen Lernfortschritt achten	Selbsttests
Organisieren	Informationsbeschaffung	Andere Informationsquellen, Medien oder Aufgaben auswählen	Suche, Schlagworte, ähnliche Videos
	externe Lernbedingungen	Lernumgebung konfigurieren	–
	Sozialform	allein, in einer Kleingruppe oder offen kollaborativ	(siehe nachfolgender Abschnitt)
Planen	Inhalte	Abschnitte und Reihenfolge des Lernweges	Playlists
	Lernziel	Lernziele bestimmen und/oder festlegen, Aufforderung zur schriftlichen Fixierung der Ziele	–
	Zeit	Lernen zeitlich planen	–

**Unterstützung durch Videolernumgebung** Unabhängig von den individuellen Fähigkeiten können externe Bedingungen das selbstgesteuerte Lernen unterstützen. Insbesondere virtuelle Lernumgebungen können Lernenden Möglichkeiten und Chancen zum selbstgesteuerten Lernen einräumen. Die Frage danach, wie eine Lernumgebung beschaffen sein muss, damit sie selbstgesteuerte Lernprozesse anregen, unterstützen und fördern kann, lässt sich jedoch ge-



genwärtig in Bezug auf das Lernen mit Videos nicht beantworten. In der Literatur konnten keine Analysen zur Akzeptanz und Nutzungshäufigkeit von Funktionen in frei zugänglichen Videolernumgebungen identifiziert werden. Mit Blick auf *YouTube*, den mit Abstand führenden Anbieter von Lernvideos im Kontext des informellen Lernens, ergeben sich einige Anhaltspunkte. Ohne an dieser Stelle der in einem folgenden Kapitel behandelten Inhaltsanalyse von Videolernumgebungen voraus zu eilen, wird in Tab. 2.2 der gegenwärtige Funktionsumfang von *YouTube* auf Konzepte des selbstgesteuerten Lernen abgebildet. Es wird sich im Verlauf dieses Buches noch zeigen, dass viele vergleichbare Videolernumgebungen ihren Anwendern eine ähnliche Unterstützung gewähren. Ungeachtet dessen, impliziert die obige Definition eine freie Wahl der Lernressourcen und Lernumgebungen, weshalb Videos und Videolernumgebungen für den Lernenden nur eine Option unter vielen anderen Lernmedien und Anwendungen darstellen.

### 2.4.3 Kollaboratives Lernen

Kollaboratives Lernen wird oft definiert als: „[...] a situation in which two or more people learn or attempt to learn something together.“ [Dillenbourg 1999]. Die Formulierung ist jedoch unpräzise und lässt viele Auslegungen (siehe Tab. 2.3) zu. Aus diesem Grund bedarf es im Rahmen dieser Arbeit einer genaueren Definition:

**Definition 2.** *Kollaboratives Lernen meint im Kontext dieser Arbeit eine computervermittelte, asynchrone Form des arbeitsteiligen oder gemeinsamen Lernens in Gruppen von mindestens zwei bis maximal 10 Personen, die Lernmaterialien studieren und Aufgaben oder Problemstellungen bearbeiten.*

**Tabelle 2.3:** Exemplarische Ausprägungen des kollaborativen Lernens [Dillenbourg 1999]

Definition	Exemplarische Ausprägungen
<i>two or more</i>	<ul style="list-style-type: none"> <li>– ein Paar</li> <li>– eine kleine Gruppe</li> <li>– ein Kurs</li> <li>– eine Gemeinschaft</li> </ul>
<i>learning something</i>	<ul style="list-style-type: none"> <li>– nehmen an Kurs teil</li> <li>– studieren Lernmaterialien</li> <li>– lösen Lernaufgabe oder Problem</li> <li>– lernen in der Praxis</li> </ul>
<i>together</i>	<ul style="list-style-type: none"> <li>– face2face oder computervermittelt</li> <li>– synchron oder asynchron</li> <li>– einmal oder mehrmals</li> <li>– arbeitsteilig oder gemeinsam</li> </ul>



Im Idealfall bilden die Lernenden in den Kleingruppen über eine gewisse Zeit hinweg eine Gemeinschaft, d.h. die Beteiligten kennen sich untereinander; sie kommunizieren und interagieren miteinander; sie verfolgen ein gemeinsames Ziel; entwickeln ein „Wir-Gefühl“; sie grenzen sich von anderen Gruppen ab und organisieren sich durch Rollen und Aufgabenzuweisungen selbst [Kerres 2012, S. 168].

Die im Rahmen dieser Arbeit betrachteten kollaborativen Lernaktivitäten beziehen sich auf „video cocreation“ und „video conversation about the video“ [Pea und Lindgren 2008], nicht jedoch auf die videovermittelte Kommunikation in Videokonferenzen (siehe dazu Ertl [2003]; Reiserer et al. [2002]). „Video cocreation“ umfasst gemeinsame Aktivitäten zur Manipulation und Ergänzung des Videomaterials einschließlich der damit assoziierten Medien (z.B. Präsentationsfolien, Kapitelmarken oder Kommentare) und hypermedialen Arrangements (z.B. Hyperlinks, Kategorisierungen). Eine Konversation anhand des Videos kann beispielsweise durch Diskussionsbeiträge, Hervorhebungen und Notizen, die mit allen Gruppenmitgliedern geteilt werden entstehen. Gegenstand der Konversation ist dabei der Inhalt des Videos bzw. die mit Hilfe des Videos zu bearbeitende Aufgabe.

In der Literatur finden sich unterschiedliche Ausprägungen kollaborativer Lernenaktivitäten anhand von Videos. In der nachfolgenden Auflistung sind diese kategorisiert, erläutert und durch mindestens eine Quelle belegt.

- **Analyse:** Im Gegensatz zu einer separaten, verschriftlichten Analyse eines Films oder einer Videoaufzeichnung stellt die zeitliche Verankerung der Analyse in Form von Annotationen einen direkten Bezug zum Inhalt des audiovisuellen Mediums dar [Reinhardt et al. 2011].
- **Reflexion:** Das von Vohle [2013] eingeführte Verfahren ermöglicht es einem Lernenden seine videografierte Lernleistung (z.B. einen Vortrag) zunächst selbst durch Textkommentare und gegebenenfalls grafische Hervorhebungen im Video zu reflektieren. Diese Kommentare können andere Lernende erwidern und ergänzen, bevor der Lehrende ein abschließendes Feedback dazu abgibt.
- **Diskussion:** Lauer und Trahasch [2005b] und Trahasch [2006] haben strukturierte Diskussionen zeitlich wie räumlich in Vorlesungsaufzeichnungen verankert. Die Strukturierung erfolgte, indem neue Diskussionsbeiträge durch Angabe eines Typs („Frage, Antwort, Erklärung, Vorschlag, Klarstellung“) kategorisiert werden mussten. Je nach ausgewähltem Beitragstyp können Satzanfänge („Prompts“) als Hilfestellung vorgegeben sein.
- **Selektieren:** Durch das Herauslösen von zeitlich und räumlich begrenzten Teilen eines Videos entstehen neue Videos, die Lernende ihren Peers zur Diskussion stellen können. Lernende können somit die Aufmerksamkeit anderer auf bestimmte Details lenken („guided noticing“), um eine gemeinsame Diskussionsgrundlage zu schaffen. Siehe dazu [Pea 2006; Zahn et al. 2010, 2012].
- **Peer Assessment:** Lernende annotieren in Kleingruppen je ein Video, indem sie Überprüfungsfragen (z.B. *Multiple-Choice*- oder Freitextaufgaben) zeitlich darin verankern. In einem zweiten Schritt wird das mit Fragen angereicherte Video einer anderen Gruppe oder den übrigen Gruppenmitgliedern zur Verfügung gestellt. Diese ist nun angehalten

die Fragen zu beantworten. Anhand der Qualität der Antworten lässt sich beispielsweise ermitteln, wie angemessen eine Frage war. Siehe dazu [Seidel 2013, 2014d].

- **Wiki:** Als Ergänzung der kollaborativen Texterstellung und -bearbeitung in Wiki-Systemen ermöglicht *Vi-Wiki* die zeitliche und räumliche Verankerung von Hyperlinks innerhalb eines Videos, um somit die Videos mit dem Hypertext der Wiki-Artikel in Beziehung zu bringen [Seidel 2012b].
- **Peer Annotation:** Lernende annotieren zusätzliche Informationen, die den Lerngegenstand inhaltlich erweitern oder ihn durch strukturgebende Elemente wie z.B. Gliederungen, Verweise oder Übersetzungen besser nutzbar machen [Seidel 2013].

**Vorzüge** Eine kollaborative Lernform ermöglicht den Lernenden ihre Sichtweisen und Perspektiven auf einen videografischen Lerngegenstand miteinander zu teilen. Sie können somit ihr Wissen (zeitlich präzise) zusammentragen und iterativ innerhalb der Lernumgebung weiterentwickeln. Dafür vorgesehene Systeme fokussieren die Interaktion mit dem System und soziale Interaktion in gleicher Weise.

In der Literatur finden sich bislang keine Hinweise auf vergleichende empirische Untersuchungen von kollaborativen und individuellen Lernformen im Zusammenspiel mit Videos. Aussagen über Auswirkungen auf den Lernerfolg, die Lernzeit, das Sozialverhalten und die Selbsteinschätzung der Beteiligten, wie sie beispielsweise Breuer [2000] medienunabhängig formulierte, sind somit weder belegt oder widerlegt.

**Limitationen** Ein hoher Freiheitsgrad in kollaborativen Lernprozessen ist laut Weinberger und Fischer [2012] jedoch auch mit einigen Herausforderungen verbunden, die auch negative Auswirkungen auf die Qualität der Arbeitsergebnisse und den individuellen Lernerfolg haben können:

- Selbstorganisation- und regulation: Wahrnehmung von Rollen und Aktivitäten, Einhaltung von Terminen und Zeitplänen.
- Schnelle Konsensbildung, d.h. Lösungsvorschläge werden ohne Abwägung von Vor- und Nachteilen oder einer Diskussion von Alternativen von den Gruppenmitgliedern angenommen.
- Bestätigungstendenz: Lernende schenken Informationen, die nicht ihrem eigenen Standpunkt entsprechen, weniger Aufmerksamkeit.
- Ungleiche Beteiligung: Leistungsstarke Lernende partizipieren übermäßig bei der Bearbeitung von Lernaufgaben

Falls Lernende nur über wenig Erfahrungen darüber verfügen, wie man in einer Gruppe effektiv zusammen arbeiten kann, ist deshalb eine Steuerung von Außen sinnvoll. Zahn et al. [2012] differenzierte für kollaborative Gestaltungsaufgaben zwei Formen der Steuerung („guidance“): Erstens, die Steuerung in Bezug auf die kognitive Herausforderung bei der individuellen Problemlösung (etwa durch Aufgabenschemata); und zweitens Steuerung der Gruppenkoordination und -kommunikation. CSCL-Scripts bieten hierzu verschiedene Lösungsmöglichkeiten und werden im Kapitel 5 (S. 171) näher erläutert. Darüber hinaus können auch Tutoren die Zusammenarbeit fördern und lenken (siehe z.B. Jödicke und Teich [2015]).

**Unterstützung durch Videolernumgebung** Reinmann [2010, S.115] hebt hervor, dass kollaborative Technologien darauf abzielen, „dass sich Lernende virtuell austauschen und zusammenarbeiten.“ Häufig angeführt werden in diesem Zusammenhang virtuelle Klassenräume, Foren oder Wikis sowie kollaborative Schreibwerkzeuge (z.B. *Etherpad*). Mit Systemen, die eine Zusammenarbeit anhand von Videos ermöglichen, gibt es bislang nur vereinzelte Berichte. Die folgende Aufstellung gibt einen Überblick über die von mehreren Forschergruppen identifizierten soziotechnischen Herausforderung bei der Gestaltung von kollaborativen Videolernanwendungen.

- **Referenzierung:** Die zeitabhängige Darstellung von audiovisuellen Informationen bedingt während der Videowiedergabe eine gewissen Flüchtigkeit der Informationen. Analysen, Reflexionen, Verknüpfungen und Diskussionen müssen einen zeitlichen und räumlichen Bezug zum audiovisuellen Medium haben. [Pea und Lindgren 2008]
- **Aufmerksamkeitslenkung:** Bei inhaltsbezogenen kollaborativen Aktivitäten sollten die beteiligten Teilnehmer wissen, welcher Inhalt gerade im Fokus steht. Lernende sollten deshalb Möglichkeiten haben, die Aufmerksamkeit ihrer Peers augenblicklich koordinieren zu können (vgl. „Simultaneity“ [Clark und Brennan 1991]). Siehe auch [Pea und Lindgren 2008].
- **Wiederauffinden:** Effektives Suchen, Wiederfinden und Nutzen von kollaborativen Arbeitsergebnissen. Siehe auch [Hofmann et al. 2009; Pea und Lindgren 2008].
- **Zugriffsrechte:** Abgesehen von sensiblen personenbezogenen und urheberrechtlich relevanten Daten, sollten Teilnehmer auch über einen persönlichen Bereich verfügen, den nur sie einsehen können. Siehe auch [Pea und Lindgren 2008].
- **Kollektive Wissensgenerierung:** Einbeziehung der in einer Gruppe gewonnenen Erkenntnisse sowie der Erfahrungen und Expertise der Gruppenmitglieder. Siehe auch [Pea und Lindgren 2008].
- **Gewärtigkeit:** Durch die zeitlich und räumlich getrennte Zusammenarbeit muss den Beteiligten ersichtlich sein, mit wem sie aktuell zusammenarbeiten können (Online-Status) und zu welcher Zeit andere Gruppenmitglieder Beiträge erbracht haben (vgl. „Reviewability“/„Revisability“ [Clark und Brennan 1991]). Siehe auch [Hofmann et al. 2009; Seidel 2014d].
- **Ergebnisdarstellung in Echtzeit:** Wird ein soeben erbrachter Beitrag eines Gruppenmitglieds nicht augenblicklich den anderen aktiven Mitgliedern der Gruppe angezeigt, besteht die Gefahr, dass Beiträge oder Lösungen mehrfach erbracht werden. Um Lernende nicht auf diese Weise zu frustrieren, sollten Beiträge idealerweise bereits während ihres Entstehungsprozesses, zumindest jedoch sofort nach der Fertigstellung für alle sichtbar werden. Clark und Brennan [1991] bezeichnen diese Eigenschaft des *Groundings* als „Contemporality“. Bedeutsam ist dies insbesondere zur Aufrechterhaltung kohärenter Diskussionen mit mehreren Teilnehmern [Pea und Lindgren 2008]. Siehe auch Seidel [2014d]; Trahasch [2006].

Im zurückliegenden Abschnitt wurden drei typische E-Learning-gestützte Lernformen hinsichtlich der Nutzung von Lernvideos betrachtet: (1) rezeptives Lernen, (2) selbstgesteuertes Lernen und (3) das kollaborative Lernen. Aus didaktischer Sicht wurde der Versuch unternommen,

Vorzüge und Limitationen der einzelnen Lernformen aus Sicht der Lernenden und Lehrenden darzustellen. Mangels ausreichender empirischer Belege gelten diese Aussagen jedoch noch nicht als gesichert, da entsprechende Erkenntnisse zumeist aus den Erfahrungen mit einem bestimmten Werkzeug abgeleitet wurden. Nichtsdestotrotz ließen sich Hypothesen für Anforderungen für Videolernumgebungen gewinnen, welche unterstützenden Technologien Lernenden bei ihren Lernaktivitäten von Nutzen sein können.

### **Zusammenfassung**

In diesem einführenden Kapitel wurde der Kontext von Lernvideos anhand der unterschiedlichen Einsatzfelder und Potentiale des Mediums erläutert. Dabei wurde deutlich, dass Lernvideos als integrales oder ergänzendes Material zur Wissensvermittlung eingesetzt werden, jedoch auch als Produkt einer Lernleistung fungieren können. Neben diesen im engeren Sinne didaktischen Anwendungen, sind Videos ein Mittel, um wissenschaftliche Arbeiten darzustellen und Studien- oder Arbeitsangebote in Bildungseinrichtungen zu bewerben.

Die aus den kognitiven Lerntheorien hervorgegangenen Gestaltungsprinzipien wurden im zweiten Abschnitt des Kapitels dargestellt und in Bezug auf Lernvideos erörtert. Sofern diese Prinzipien im Gestaltungs- und Produktionsprozess berücksichtigt werden, können sie dazu beitragen die kognitive Verarbeitung der audiovisuellen Lerninhalten zu erleichtern und kognitive Belastungen zu reduzieren. Die Notwendigkeit zur Anwendung dieser Prinzipien ist teilweise von den Zielgruppen abhängig, die beispielsweise über unterschiedliches Vorwissen und Sprachkenntnisse verfügen. Dabei ist festzustellen, dass die Umsetzung dieser Prinzipien nicht nur bei der Videoproduktion, sondern teilweise auch noch vor oder während der Auslieferung im Videoplayer erfolgen kann. Die Prinzipien geben jedoch noch keine Auskunft darüber, wie sie technisch und gestalterisch im *User Interface* einer Videolernumgebung realisiert werden können.

Im dritten Abschnitt wurde Schulmeisters Taxonomie der Interaktivität [Schulmeister 2005] auf Lernvideos projiziert und anhand von Beispielen verdeutlicht. An dieser Stelle deuten sich bereits Querbeziehungen zu Hypervideos bzw. interaktiven Videos an, die sich in ähnlicher Weise wie Hypertexte im Web bzw. *Web 2.0* rezipieren, manipulieren und adaptiv nutzen lassen. Für die weitere Betrachtung dieser Aspekte unterscheidet man Mikro- und Makrointeraktivität, d.h. die Interaktivität innerhalb eines Videos und innerhalb des medialen Gesamtarrangements.

Im letzten Abschnitt wurden rezeptive, selbstgesteuerte und kollaborative Lernformen in Bezug auf die Nutzung von Lernvideos betrachtet, die sich in verschiedenen Lernszenarien wiederfinden. Für die Identifizierung von Design Patterns wie auch für die Entwicklung von adäquaten Videolernumgebungen sind die in diesem Abschnitt ermittelten Gestaltungsanforderungen relevant. Dies gilt im Besonderen für kollaborative Lernumgebungen, in denen die Interaktion von Mitgliedern einer Gruppe durch ein Script gesteuert wird.

**Kernpunkte**

- Videoplayer bzw. Lernumgebungen können dazu beitragen, die kognitive Verarbeitung bestehender videografischer Informationen zu unterstützen und die kognitive Last zu reduzieren.
- Die Unterscheidung zwischen Mikro- und Makrointeraktivität ist auf technischer Ebene durch den Videoplayer und die Videolernumgebung repräsentiert.
- In Bezug auf das Lernen mit Videos gibt es vier wesentliche Lernformen: rezeptives Lernen, selbstgesteuerte Lernen, kollaboratives Lernen und Lernen in Co-Präsenz, wobei nur die ersten drei im E-Learning eine Rolle spielen.
- Kollaboration anhand von Videos kann sich auf die Analyse, Reflexion, Diskussion und Selektion sowie auf spezielle Formen des Peer Assessments und der Peer Annotation im Sinne der Wissensgenerierung beziehen.
- Für das selbstgesteuerte und kollaborative Lernen gibt es nur vereinzelte Software Lösungen zur Unterstützung des Lernens.



## Kapitel 3

# Videolernumgebungen und Hypervideos

In diesem Kapitel stehen technische Aspekten von Videolernumgebungen und Hypervideos im Vordergrund. Das übergeordnete Ziel besteht darin, technische Anforderungen für die Realisierung von Videolernumgebungen festzulegen. Zu Beginn werden grundlegende Begrifflichkeiten wie Video und Videolernumgebung sowie Hypervideo und interaktives Video im Kontext verschiedener Formen von Hypermedia definiert und eingeordnet. Es wird sich dabei zeigen, dass Videolernumgebungen eine spezielle Form von Hypervideo-Anwendungen darstellen. Um bestimmte Erscheinungsformen und Verhaltensweisen von Hypervideos besser erklären zu können, ist es hilfreich auf etablierte Modelle und Beschreibungsformate zurückzugreifen. Im weiteren Verlauf dieser Arbeit wird auf diese grundlegenden Konzepte aufgebaut.

Im zweiten Abschnitt des Kapitels steht das *User Interface* von Videolernumgebungen im Mittelpunkt. Ziel ist es, die Besonderheiten dieser zeitlich determinierten Schnittstelle zwischen Mensch und Video herauszuarbeiten und den Gestaltungsraum videobasierter Lernumgebungen zu beschreiben. Auf der Grundlage von Literatur werden Herausforderungen einer benutzerfreundlichen Gestaltung des *User Interfaces* identifiziert.

Im Hinblick auf die Realisierung von Videolernumgebungen mit Hilfe von Interaction Design Patterns und unter Einbeziehung von CSCL-Scripts ist es notwendig, die technischen Rahmenbedingungen der Implementierung im Web zu berücksichtigen. Der dritte Abschnitt lenkt deshalb das Augenmerk auf wesentliche technische Herausforderungen. Typisch für Hypervideos ist dabei die *on demand* Verfügbarkeit der Medien. In folgedessen kann das Datenaufkommen steigen und im ungünstigen Fall die Betrachtung verzögern. Eine weitere Herausforderung betrifft die akkurate Synchronisation von Videos und zeitabhängigen Annotationen mit Hilfe von Web-Technologien. Eine weitere Herausforderung ergibt sich aus der Besonderheit audiovisueller Medien, zeitlich verankerte Informationen erst im Verlauf der Wiedergabe preiszugeben. Um Informationseinheiten gezielt selektieren und aufrufen zu können, ist es notwendig die entsprechenden Zeitpunkte oder Medienfragmente exakt zu adressieren. Die technischen Rahmenbedingungen stellen wichtige Randbedingungen und Anforderungen für das in Abschnitt 6.1 vorgestellte Framework für interaktive Videos dar.

Im darauffolgenden, vierten Abschnitt geht es um praktische Ansätze zur Erstellung von Hypervideos sowie um Möglichkeiten selbige in bestehende Lernumgebungen einzubinden. Die Produktion von Hypervideos ist ein wichtiger Aspekt der Modellierung und Durchführung von CSCL-Scripts. Typische Einsatzszenarien umfassen die Aufbereitung (z.B. Selektion von Videos und Szenen, Einbettung in Lernressourcen) und Anreicherung mit Zusatzinformationen (z.B. Metadaten, synchrone Medien). Ausgehend von typischen operativen und organisatorischen Vorgehensweisen in der Post-Produktion von Hypervideos werden verbreitete Paradigmen von Autorenumgebungen erläutert. Technische Lösungsansätze für die Distribution und Integration von hypervideobasierten Lernressourcen sind im zweiten Teil des Abschnitts kurz dargestellt und durch Beispiele belegt. Die Lösungsansätze betreffen Schnittstellen für Lerninhalte und Teilanwendungen sowie Austauschformate und semantische Auszeichnungen zum Wiederauffinden von Ressourcen.

## Terminologie und Modelle

### 3.1.1 Videolernumgebung

Bevor auf die Gestaltung, Konzeption und Entwicklung von Videolernumgebungen eingegangen wird, ist es zunächst erforderlich diesen Typus einer computerbasierten Lernanwendung genauer zu spezifizieren. In der Literatur und Praxis sind diese Anwendungen unter dem Oberbegriff *Learning Management System* (LMS) zusammengefasst. Der Kern eines LMS besteht aus einer administrierbaren Datenbank mit Lernressourcen, die für eine Zielgruppe zum Lernen verfügbar sind und die Anwender auch beim Lernen unterstützt. In diesem Kapitel sollen videobasierte LMS respektive Videolernumgebungen als eine spezielle Unterkategorie der LMS im Mittelpunkt der Betrachtung stehen. Die folgenden, von Seel und Ifenthaler [2009, S. 172-173] genannten Hauptfunktionen von LMS werden daher als bekannt vorausgesetzt und nicht näher betrachtet: „das Management von Benutzerprofilen (Autoren, Lernende, Lehrende), die Administration von Kursdaten (Inhalt, Termine), und Benutzern (Rechtevergabe), die Bereitstellung von Autorenwerkzeugen zur Erstellung von Kursen, Lerninhalten, Lernaufgaben, den Einsatz von synchronen (Chat) und asynchronen (Foren, E-Mail) Kommunikationstools sowie den Einsatz von Evaluations- und Diagnosetools“.

**Definition 3.** *Eine Videolernumgebung zeichnet sich gegenüber einem herkömmlichen LMS durch drei Merkmale aus: Erstens stellen Videos die primären Lernressourcen dar; zweitens erfahren Lernende eine Unterstützung bei der Nutzung der Lernvideos; und drittens stehen Autorenwerkzeuge bereit, mit denen die zeitkontinuierlichen Medien in ähnlicher Weise editierbar sind, wie Hypertext-Inhalte im WWW.*

Damit ist nicht ausgeschlossen, dass auch herkömmliche LMS diese Merkmale aufweisen können. Für die derzeit weit verbreiteten LMS trifft dies jedoch nicht zu. Tabelle 3.1 enthält eine Analyse dieser LMS hinsichtlich ihrer Funktionen zur Lernunterstützung beim Lernen mit Videos und den Bearbeitungswerkzeugen für Autoren. Die Funktionalitäten beschränken sich im Wesentlichen auf die in gängigen Web-Browsern umgesetzten HTML5-Spezifikationen. Videos werden dabei als in sich geschlossene Entitäten ausgeliefert und dargestellt.



**Tabelle 3.1:** Übersicht über LMS und ihre Funktionalität in Bezug auf Lernvideos

Name	Videofunktionalität	Distribution, URL
<i>Blackboard Learn</i>	Videos lassen sich als Datei-Ressource hochladen, einbetten und wiedergeben. Videos von ausgewählten Fernsehsendern (z.B. NBC) lassen sich als Feed oder als einzelnes Video in einen Kurs einbetten.	kommerziell, <a href="http://www.blackboard.com/">http://www.blackboard.com/</a>
<i>IMC Learning Suite (Clix)</i>	Videos können einem Kurs als Datei hinzugefügt werden. Außerdem lassen sie sich in (HTML-)Folien integrieren oder mit Folien synchronisieren.	Open Source, <a href="http://www.clix.de/">http://www.clix.de/</a>
<i>ILIAS</i>	Videos kann man als Datei-Ressource hochladen, einbetten und wiedergeben. .	Open Source, <a href="http://www.ilias.de/">http://www.ilias.de/</a>
<i>moodle</i>	Videos lassen sich als Datei-Ressource hochladen oder im mitgelieferten <i>Flowplayer</i> sowie im Player eines externen Dienstes (z.B. von <i>YouTube</i> ) einbetten. Darüber hinaus gibt es eine Vielzahl von Plugins, die den Videoeinsatz unterstützen.	Open Source, <a href="http://www.moodle.org/">http://www.moodle.org/</a>
<i>OLAT</i>	Video-Podcasts lassen sich im Kursbaustein Podcast durch Angabe einer <i>Feed-URL</i> abonnieren (z.B. von <i>iTunes U</i> ) oder als Video hochladen. Lernende können die Podcasts je Datei bewerten und kommentieren. Eine Einbettung in HTML-Seiten ist mit Hilfe von <i>IFrames</i> möglich. Auch innerhalb von SCORM-Paketen ist eine Einbindung möglich.	Open Source, <a href="http://www.olat.org/">http://www.olat.org/</a>
<i>Stud.IP</i>	Als Datei können Videos in einen Kurs eingefügt werden. Sie sind durch einfache Befehle in den Lernmaterialien referenzierbar und können für die Wiedergabe in einem Player eingebunden werden. Darüber können Videos auch Bestandteil des Forums/Chats, genannt <i>Blubber</i> , darstellen. Mittels Plugins können Videos von <i>YouTube</i> oder <i>Vimeo</i> eingebunden werden.	Open Source, <a href="http://www.studip.de/">http://www.studip.de/</a>

Die eingangs genannte Forderung nach Lernunterstützung enthält eine bemerkenswerte Implikation. Lernende können nur dann durch das umgebende System eine Unterstützung erfahren, wenn dieses Interaktionen mit dem System und indirekt mit dem Lerngegenstand und anderen Personen (Lernende, Lehrende) ermöglicht. Wie bereits im vorherigen Kapitel festgestellt wurde, sind moderne Mensch-Computer-Schnittstellen und somit auch LMS per Definition interaktiv (vgl. S. 44). Dies schließt die darin eingebundenen Lernvideos mit ein, weshalb in Bezug auf Videolernumgebungen immer von interaktiven Videos oder Hypervideos auszugehen ist.

### 3.1.2 Terminologie: Video, Hypervideo und interaktives Video

Mit Blick auf die Definition von Lernvideos (S. 30) ist die technische Beschreibung eines bewegten Bildes umfassender, als es typische Dateiformate suggerieren. Als Video bezeichnet man

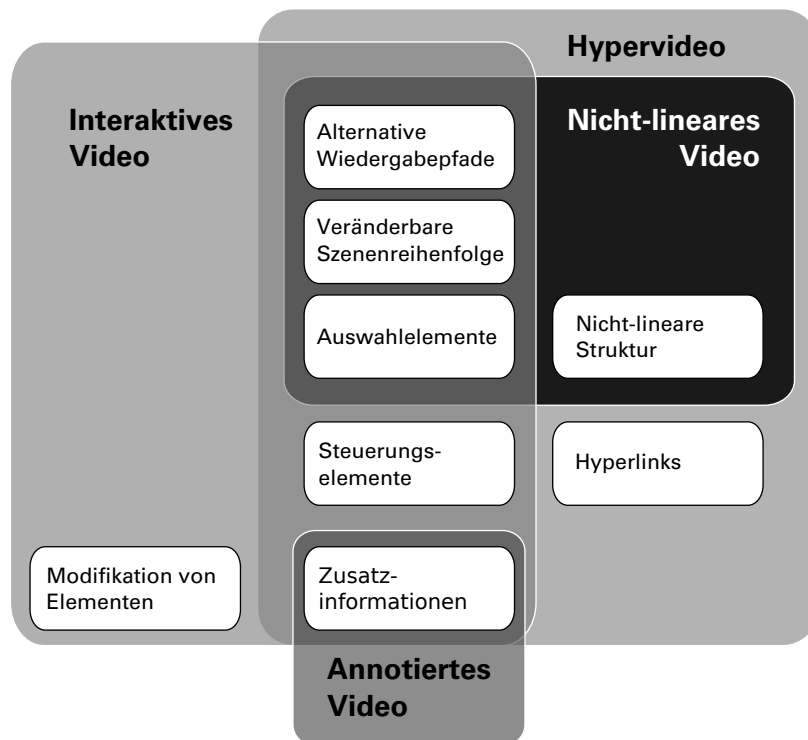
ein kontinuierliches Medium, welches mindestens eine visuelle, idealerweise auch eine akustische Dynamik in seiner Darstellung aufweist und vom menschlichen Subjekt als Betrachter wahrnehmbar ist. Die Besonderheit dieser Definition besteht in der bedingten und impliziten Einordnung als Multimedia, indem Bild mit Ton kombiniert sind. Aus technischer Sicht müssen Ton-, Bild- bzw. Bewegtbilddaten nicht in einer Datei zusammengefasst vorliegen, so lange sie in der Multimediapräsentation zusammengeführt werden. Wie in *photofilms*, *web documentaries* oder *web comics* üblich, kann ein Video auch aus einer animierten Folge von Einzelbildern bestehen, die durchgehend von einem Sprecher begleitet werden [Westbrook 2013]. Die Einschränkung auf die Wahrnehmbarkeit der Darstellung ist notwendig, da insbesondere zu schnelle Bildwechsel, mangelnde Kontraste bzw. zu tiefe oder zu hohe Tonfrequenzen für den Anwender keine verarbeitbare Informationen liefern.

Zur Definition von Hypervideo ist es zunächst dienlich den Oberbegriff Hypermedia als Schnittmenge von Hypertext und Multimedia in Anbetracht ihrer Präsentation und Manifestation in Dokumenten einzuführen:

„Hypertext is a description of the referencing information among self-contained units of information. A hypertext document is a collection of information units and referencing information, called links [. . .]. A hypertext presentation is the runtime manifestation of one or more hypertext documents with which a reader can interact allowing the reader to navigate through the hypertext document. The information units in a hypertext document may include media types other than text. A commonly used term for this is hypermedia. Multimedia is a collection of multiple units of information that are constrained by temporal synchronization relationships. A multimedia document is a collection of information units and associated synchronization information [. . .]. A multimedia presentation is the runtime manifestation of a multimedia document. A reader can interact with a multimedia presentation by, e.g., starting or pausing the presentation. [. . .] A hypermedia document is thus a collection of multimedia documents along with referencing information. A hypermedia presentation is the runtime manifestation of one or more hypermedia documents. A reader can interact with a hypermedia presentation either as a multimedia presentation, by starting or pausing a multimedia presentation, or as a hypertext presentation, by navigating through the information units.“ [Hardman 2004, S. 10f]

Hypermedia kann man anhand der Medientypen differenzieren, die im jeweiligen Dokument die tragende Rolle spielen und die Repräsentation dominieren bzw. steuern. Hypertext und *Hypergraphic* dominieren durch die räumliche Darstellung von Text bzw. Bild, während bei *Hyperaudio* und Hypervideo die kontinuierlichen Medien Audio bzw. Video die Darstellung der übrigen Objekte zeitabhängig steuern. Zur genaueren Eingrenzung des Begriffs Hypervideo genügt die obige Definition jedoch noch nicht. Es ist genauer zu klären, durch welche Eigenschaften sich Hypervideo-Präsentationen auszeichnen.

Für videobasierte interaktive Systeme finden sich in der Literatur eine Reihe von Bezeichnungen: *non-linear video*, *interactive video*, *hypervideo*, *annotated video* und vereinzelt ist auch von *layered video*, *video 2.0* und *temporal web* die Rede. In der Forschung besteht jedoch kein



**Abbildung 3.1:** Merkmale verschiedener Formen von nicht-linearen Videos im Überblick (Abb. in Anlehnung an Meixner et al. [2012])

begrifflicher Konsens. Meixner et al. [2012] haben Begriffsdefinitionen aus 25 Publikationen. Die Kategorien und Eigenschaften, welche sie für den Vergleich hinzugezogen haben, sind unten stehend gelistet. Hypervideo und interaktive Videos kann man demnach als Oberbegriffe für *non-linear* und *annotated video* ansehen, wobei interaktive Videos gegenüber Hypervideos auch die Modifikation von Inhalten vorsehen (vgl. Abb. 3.1). In wie weit die Navigation interaktiver Videos, technisch wie gestalterisch, wirklich ohne Hyperlinks auskommen kann, muss bezweifelt werden. Aus diesem Grund werden die beiden Begriffe als synonyme Bezeichnungen verwendet. Sie definieren damit Anwendungen mit folgenden Eigenschaften:

- **Steuerelemente:** Während der Szenenverlauf unverändert bleibt, kann der Betrachter die Wiedergabegeschwindigkeit und die Bildqualität steuern [Meixner et al. 2012]. In einigen Fällen sind auch Kameraposition, Kamerabewegung und der Winkel der Kamera justierbar; man bezeichnet dies als cinematografische Navigation [Vardi 1999]; z.B. in DIVER™ [Pea 2006].
- **Nicht-lineare Struktur:** Ohne festgelegte Wiedergabereihenfolge sind die Videoszenen in Form eines Baums oder Graphen strukturiert und entlang dieser Pfade bzw. Kanten navigierbar [Lehner und Siegel 2009; Meixner et al. 2012].
- **Alternative Wiedergabepfade:** Innerhalb eines linearen Videos kann man auf festgelegte zeitliche Positionen springen. Die Positionen sind auf der Zeitleiste markiert [Meixner et al. 2012].
- **Änderung der Szenenabfolge:** Der Betrachter kann Szenen wahlfrei zur Wiedergabe auswählen ohne deren strukturelle Beziehungen kennen zu müssen [Meixner et al. 2012].

- **Hyperlinks:** Verweise auf interne und externe Ressourcen, wie andere Videos, Videoszenen oder Webseiten [Meixner et al. 2012].
- **Zusatzinformationen:** Neben dem Video oder innerhalb des Videobildes können diskrete oder kontinuierliche Medien erscheinen (z.B. Bild oder Text bzw. Video oder Audio) [Lehner und Siegel 2009; Meixner et al. 2012].
- **Modifikation/Ergänzung von Inhalten:** Der Betrachter ist in der Lage das Video sowie die angereicherten Informationen zu modifizieren und neue Informationen hinzuzufügen [Meixner et al. 2012].
- **Auswahlelemente:** Der Aufruf weiterer Szenen erfolgt durch Aktivierung sensitiver Flächen, *buttons* oder die Beantwortung von Quizfragen [Lehner und Siegel 2009; Meixner et al. 2012]. Letztere Form bezeichnet man insbesondere als *adaptive hypervideo*.

**Tabelle 3.2:** Die sechs Stufen der Interaktivität nach Schulmeister [2005, S. 13] in Relation zu den Hypervideo-Eigenschaften von Meixner et al. [2012]

	1. Objekte betrachten	2. Multiple Darstellungen	3. Variation der Repräsentationsform	4. Inhalt der Komponente beeinflussen	5. Objekte und Prozesse generieren	6. Situationsabhängiges Feedback
Steuerungselemente	•					
Nicht-lineare Struktur	•	•				
Alternative Wiedergabepfade	•	•				
Änderung der Szenenabfolge	•	•				
Hyperlinks	•	•				
Zusatzinformationen		•				
Modifikation/Ergänzung von Inhalten			•	•		
Auswahlelemente					•	

Die genannten Eigenschaften lassen sich, wie in Tab. 3.2 dargestellt, den in Abschnitt 2.3 erwähnten sechs Interaktivitätsstufen von Lernemedien zuordnen (siehe S. 44). Die Stufen 2 und 3 – die Wahl multipler Darstellungsformen und die Variation der Repräsentationsform durch den Benutzer – erfahren dabei in den von Meixner et al. [2012] betrachteten Forschungsarbeiten eine größere Aufmerksamkeit.

Ausgehend von der Definition und den Eigenschaften von Hypervideos, stellt sich die Frage der technischen Realisierung. Es ist offensichtlich, dass ein Hypervideo nicht allein durch eine Videodatei repräsentiert werden kann. Deshalb ist es erforderlich, sich mit Hypermedia-Dokumenten vertraut zu machen. Diese manifestieren Videos und weitere Elemente mit den

genannten Eigenschaften und ermöglichen die Ausführung und Darstellung in einer Laufzeitumgebung.

### 3.1.3 Spezifikationen für Hypervideo-Dokumente

Wie oben ausgeführt, bestehen Hypermedia-Dokumente aus einer miteinander verlinkten Sammlung von Multimedia-Dokumenten, die sich ihrerseits aus zeitlich miteinander assoziierten, multimedialen Informationseinheiten zusammensetzen [Hardman 2004, S. 10-12]. In den verschiedenen Laufzeitumgebungen können diese Dokumente ad-hoc sowie mit Hilfe von multimedia- oder annotationsbasierten Sprachen repräsentiert sein.

Bei einer Reihe von Anwendungen werden Hypervideos ad-hoc spezifiziert, d.h. statisch im Programmcode verankert [Sadallah et al. 2012]. Die Formate kann man als proprietär bezeichnen, da sie im Gegensatz zu expliziten und generischen Lösungen schwer auf andere Systeme zu übertragen sind. Sie entziehen sich damit der Weiterentwicklung von Dokumentenmodellen [Sadallah et al. 2012].

Im Gegensatz dazu stützen sich multimediasbasierte Sprachen auf generische Auszeichnungssprachen. Als Beispiele sind die drei Hauptvertreter zu nennen: *Synchronized Multimedia Integration Language* (SMIL) und die auf dem *Nested Context Model* aufbauende *Nested Context Language* (NCL 3.0) [Soares 2009] sowie ferner die *Video Annotation Markup Language* [Zhou und Jin 2006]. Siehe auch Teilabschnitt 3.3.4, S. 80.

Bei annotationsbasierten Spezifikationen befinden sich annotierte Daten in einer zeitlichen Ordnung entlang einer Zeitleiste und werden in einer gesonderten Schicht auf dem tragenden Videoelement präsentiert [Halasz und Schwartz 1994]. Diese Annotationen sind mit einem Video-segment assoziiert und insofern ein Metadatum mit zeitlichem Bezug. Die Ausdrucksmächtigkeit dieser Spezifikationsform ist gegenüber den multimediasbasierten Sprachen eingeschränkt, da z.B. komplexere zeitliche Relationen (siehe auch Teilabschnitt 3.1.4, S. 66) nicht umsetzbar sind.

Um unterschiedliche Spezifikationen oder Systeme beschreiben und vergleichen zu können, ist es hilfreich eine abstraktere Beschreibungsweise zu verwenden. Dokumentmodelle bieten sich hierfür an, da sie die Spezifikationen verschiedener Dokumente sowie das Verhalten der sie verarbeitenden Systeme abstrahieren, indem sie Struktur, Timing, Layout, Interaktionen für die Spezifikation von interaktiven, zeitbasierten Online-Präsentation erfassen [Hardman 2004]. Zu den wichtigsten Modellen zählt das *Amsterdam Hypermedia Model* AHM [Hardman 2004], was aus dem *Dexter Hypertext Model* [Halasz und Schwartz 1994] hervorging, sowie das *Nested Context Model*. Eine vergleichende Darstellung der Modelle findet sich bei Meixner und Kosch [2012].

Festzuhalten bleibt, dass multimedia- oder annotationsbasierte Spezifikationen den Ad-hoc-Spezifikationen aufgrund ihrer generischen Eigenschaften überlegen sind. Eine Entscheidung zwischen multimedia- und annotationsbasierten Spezifikationen bleibt ein Kompromiss zwischen Ausdrucksmächtigkeit und Implementierbarkeit. Für die Entwicklung von Hypervideo-Systemen ist es jedenfalls wichtig, geeignete Datenstrukturen zu finden, die zeitliche wie räumliche Beziehungen zwischen Videos und den annotierten Zusatzinformationen aufweisen und

eine gewisse Interoperabilität gewährleisten. Im Teilabschnitt 3.3.4 auf S. 80 wird diese technische Herausforderung bei der Analyse von Systemanforderungen noch einmal genauer thematisiert. Dafür bedarf es jedoch einer genaueren Erläuterung bestimmter zeitlicher Modelle, die als zentrale Bestandteile von Hypermedia-Modellen sowohl für die technische Beschreibung von Systemverhalten, als auch für die Beschreibung von Dokumenten eine tragende Rolle spielen.

#### 3.1.4 Modelle des zeitlichen Layouts

Zeit ist ein wesentliches Merkmal von Multimediapräsentationen. Während statische Dokumente oder gewöhnliche Hypertext-Systeme wie HTML über ein spatales Layout verfügen, aus dem hervorgeht, wo die Objekte in einem Dokument erscheinen sollen, benötigen Multimediapräsentationen darüber hinaus ein zeitliches Layout [Buchanan und Zellweger 2005]. Das zeitliche Layout dient der Steuerung der Darstellung eines Multimediadokuments [Buchanan und Zellweger 2005]. Es besteht in der Regel aus einer Verknüpfung von Zeitpunkten oder Zeitspannen mit darstellungsrelevanten Ereignissen und ist Teil des Multimediadokuments. In diesem Teilabschnitt stehen Modelle im Mittelpunkt, mit deren Hilfe man zeitliche Layouts im Hinblick auf ihre Anwendung in Hypervideo-Playern und Autorenumgebungen beschreiben kann. Neben der Einführung von zentralen Begriffen, geht es insbesondere um formale Beschreibungsformen von Ordnungsbeziehungen zwischen zeitlichen Intervallen in Multimediadokumenten. Innerhalb dieser Arbeit sind diese Grundlagen insbesondere bei der Spezifikation videobasierter CSCL-Scripts (siehe S. 184) und bei den Datenspezifikationen für interaktive Videos im Framework *VI-TWO* (siehe S. 225) relevant.

Eingangs ist es notwendig den Begriff der Zeit im Kontext von digitalen Medien genauer zu betrachten, denn es gibt durchaus unterschiedliche Sichtweisen auf die Zeit. Hardman et al. [1999] unterscheiden die Zeit in einem Modell anhand des Medienelements und dem Arrangement im *hypermedia document* sowie anhand der idealen und tatsächlichen Darstellung bei der Wiedergabe. Von diesem Modell ausgehend lassen sich Aspekte des zeitlichen Layouts, wie die möglichen Relationen zeitlicher Intervalle in einem Dokument [Allen 1983; Duda und Keraman 1995] oder die Formen von zeitabhängigen Verlinkungen zur Dokument- und Laufzeit genauer beschreiben.

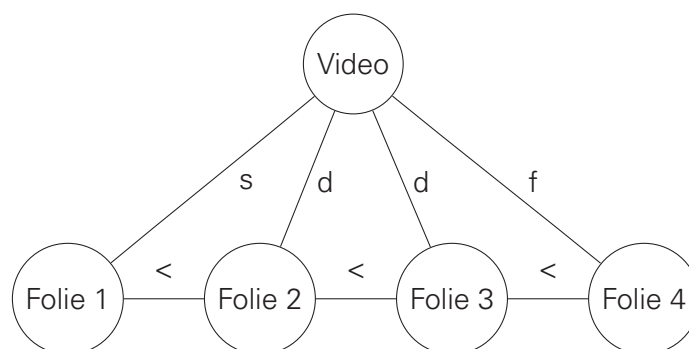
- **Medienelementzeit (*media element time*):** Gemeinhin als Wiedergabezeit bezeichnet, ist ein Bestandteil des dargestellten Mediums. Video- und Audiodateien bzw. deren Fragmente besitzen eine intrinsische Wiedergabezeit, während Bilder und Texte eine unendliche Wiedergabedauer aufweisen.
- **Dokumentzeit (*document time*):** Unabhängig von der *media element time* kann man einem Medienelement eine Darstellungszeit zuweisen, indem man es beispielsweise mehrfach wiederholt oder nur Ausschnitte davon zeigt bzw. in seiner zeitlichen Ausdehnung streckt oder staucht.
- **Darstellungszeit (*rendered time*):** Die Darstellungszeit bezeichnet die Zeit, in der die Medien eines Dokuments in einem idealen System wiedergegeben werden könnten.
- **Laufzeit (*run time*):** Entspricht der tatsächlichen Darstellungszeit einschließlich der Unterbrechungen durch Übertragungsverzögerungen im Netzwerk sowie Benutzerinteraktio-

nen auf der Zeitleiste (z.B. Play, Pause, Vor- und Rückwärtssprünge) bzw. der Navigation zwischen verschiedenen Medienrepräsentationen.

Je nach Video-Codec ist die intrinsische Wiedergabezeit (Medienelementzeit) anhand von Metadaten der Videodatei ermittelbar. Alternativ ist eine probabilistische Berechnung aus dem Verhältnis von Darstellungszeit und der bis dahin dekodierten Daten möglich. Der Darstellungszeit kommt eine untergeordnete Rolle zu, da sich die Zeitdauer der Wiedergabe durch Benutzerinteraktionen auf der Zeitleiste oder durch die Navigation von einem Video zum anderen sich beliebig verkürzen oder verlängern kann. Unabhängig davon stellt die Laufzeit aus technischer Sicht zumindest im WWW eine Herausforderung dar (vgl. Abschnitt 3.3, S. 76).

Für die Entwicklung von Hypervideo-Playern und Autorenumgebungen ist vor allem die Dokumentzeit relevant. Zum einen geht es um die Frage der Ausdrucksmächtigkeit hinsichtlich der Umsetzung der möglichen Relationen zwischen zeitlichen Intervallen, wenn beispielsweise Videoszenen mit Standbildern in eine zeitliche Beziehung gebracht werden sollen (siehe Abb. 3.2). Zum anderen werden nicht selten zeitbezogene Verweise innerhalb eines Medienelements und zwischen unterschiedlichen Elementen definiert, die dem Nutzer zur Laufzeit eine individuelle Repräsentation der Inhalte ermöglicht.






Die von Allen [1983] eingeführte *interval based temporal logic* umfasst eine grundlegende Beschreibung von Ordnungsbeziehungen zwischen zeitlichen Ausdehnungen respektive Intervallen. Ein Intervall kann ein ganzes Mediensegment, ein Teilsegment (z.B. die erste Szene in einem Video) oder auch einen Zusammenschluss mehrerer Intervalle umfassen [Buchanan und Zellweger 2005]. Zwei zeitliche Intervalle stehen durch die Relationen *before*, *equal*, *meets*, *overlaps*, *during*, *starts*, *finishes* sowie deren inverse Relationen miteinander in Beziehung. Insgesamt ergeben sich 13 Relationen, die sich gegenseitig ausschließen (siehe Tab. 3.3). Die formale Notation kommt dabei ohne genaue Zeitangaben (z.B. Zeitangaben in Sekunden, gemäß ISO 8601 [Of et al. 2000]) aus (vgl. [Stotts und Furuta 1990]). Es handelt sich um relative Beziehungen, die keine Aussagen über die Kausalität der Intervalle zulassen [Duda und Karaman 1995]. Beispielsweise geht aus der Relation *X gefolgt von Y* nicht eindeutig hervor, ob X Y startet, ob Y X beendet oder ob beide Intervalle in keinem kausalen Zusammenhang stehen. Kausale Beziehungen zwischen verschiedenen Medienelementen finden häufig in annotationsbasierten Hypermedia-Dokumenten Anwendung. Die aktuelle Wiedergabezeit eines Videos impliziert zum Beispiel die Anzeige bestimmter Zusatzinformationen.



**Abbildung 3.2:** Ein Beispiel für die zeitlichen Beziehungen zwischen einem Video und vier synchronisierten Folien



**Tabelle 3.3:** Die 13 möglichen zeitlichen Relationen zwischen zwei zeitlichen Intervallen nach Allen [1983]

Relation	Symbol	Inverses Symbol	Beispiel	Funktionale Relation
X vor Y	<	>	XXX YYY	–
X gleichzeitig zu Y	=	=		X startet Y und X beendet Y X startet Y und Y beendet X
X gefolgt von Y	m	mi	XXXYYY	X startet Y Y beendet X X ist unabhängig von Y
X überlappt Y	o	oi		–
X während Y	d	d		–
X startet mit Y	s	si		X startet Y Y startet X
X endet mit Y	f	f		X beendet Y Y beendet X

Zur Beschreibung von Medienkompositionen lassen sich die genannten Relationen auch in einen Graphen überführen. Die Knoten des Graphen bilden die einzelnen Intervalle der Medienelemente, z.B. Videoszenen oder Standbilder. Eine Kante im Graphen bezeichnet eine Relation zwischen zwei Knoten. Die Notation als Graph enthält alle Informationen darüber, wie die Intervalle miteinander in Beziehung stehen. In Abb. 3.2 sind exemplarisch die Beziehungen eines Videos zu vier synchronisierten Präsentationsfolien dargestellt. Folie 1 beginnt und Folie 4 endet zeitgleich mit dem Video. Die Folien 2 und 3 beginnen und enden während der Videowiedergabe, wobei die Reihenfolge der Folien eindeutig definiert ist.

Für einen Autor einer Komposition von kontinuierlichen Medienelementen mag die Notation oder Graphenrepräsentation zunächst einmal keine Rolle spielen. Um jedoch Autorenumgebungen, Player oder auch videobasierte CSCL-Scripts formal beschreiben zu können, bedarf es dieses Hilfsmittels.

Im zurückliegenden Abschnitt wurden bereits einige Bedien- und Steuerungselemente als Bestandteile von Hypermedia-Dokumenten erwähnt. Dabei lag das Augenmerk auf der technischen Spezifikation und den zeitlichen Relationen. Im folgenden Abschnitt werden diese Dokumente aus der Perspektive der Anwender betrachtet, die Hypervideos nutzen.

## Human Video Interface

Als Teilmenge der Schnittstelle zwischen Mensch und Computer wird im Rahmen dieser Arbeit ein *Human Video Interface* als der Teil eines Videosystems definiert, mit dem der Anwender mit dem Video kommunizieren kann. Der Anwender tritt dabei mit dem Video in Interaktion, indem er das Videosystem instruiert und von ihm Rückmeldung erhält. Unter einem Video-



system versteht man Videoinhalte, die mittels Steuerungselementen manipuliert, ergänzt und betrachtet werden können. Die Interaktionsmöglichkeiten können dabei über die basale Steuerung der Wiedergabe (Play/Pause, Timeline-Slider, Vollbild, Lautstärkeregler) hinaus gehen. Die Kommunikation zwischen Anwender und Videosystem subsumiert verschiedene Ein- und Ausgabeinformationen des Videosystems sowie die durch das System vermittelte Kommunikation mit anderen Anwendern. Diese videovermittelte Kommunikation wird vom Video getragen und durch das Video bestimmt, indem gemäß dem Master-Slave-Prinzip Objekte und Interaktionselemente in Abhängigkeit der Zeit sichtbar bzw. benutzbar werden oder einen zeitlichen Bezug zum Video aufweisen. [Seidel 2012c]

Bei der Gestaltung von Komponenten des *Human Video Interfaces* steht der durch Technologie unterstützte Anwender als menschliches Subjekt im Mittelpunkt (*user centered design*). Richtet man die Gestaltung auf den Lernenden aus (*learner centered design*), ist die Benutzerfreundlichkeit der Schnittstelle zwischen Mensch und Video eine notwendige Voraussetzung für das effiziente Lernen mit Videos. Gross [2012, S. 80] schreibt dazu: „Nur durch eine sinnvolle, das heißt möglichst kurze und nach den Benutzervorstellungen logisch erscheinende, Abfolge von Schritten durch die Softwareanwendung kann eine einfache Einarbeitung der Lernenden und Lehrenden und ein effektives und effizientes Arbeiten der Lernenden und Lehrenden mit der Software erreicht werden.“ Insbesondere der Zugang zu videografischer Information, das Design interaktiver Bedienelemente im Video sowie deren zeitliche Ausdehnung spielen hier vor dem Hintergrund der kognitiven Belastung der Anwender eine zentrale Rolle.

In Bezug auf das Lernen mit Videos existieren *Human Video Interfaces* im Kontext von technischen, sozialen und organisatorischen Gegebenheiten. Unter technischen Gegebenheiten verstehen sich beispielsweise die Datenübertragungskapazität und die Performance des Client-Systems bei der wahlfreien Wiedergabe und zeitgenauen Adressierung von Videosegmenten verschiedener Kodierungen (Video-Codecs). Von zentraler Bedeutung sind ebenso Eingabegeräte wie Maus, Tastatur, Touchscreen sowie Ausgabegeräte, i.d.R. Displays verschiedener Größe. Auch die Möglichkeiten, den Videoplayer funktional zu erweitern oder als Modul in andere Lernumgebungen zu integrieren bzw. mit anderen Diensten im WWW zu koppeln sind Teil der technischen Gegebenheiten.

Gewohnheiten der Anwender audiovisuelle Medien in Präsenz oder virtuell zu rezipieren kennzeichnen Aspekte sozialer Gegebenheiten, die bei der Gestaltung des *Human Video Interface* zu berücksichtigen sind. Der aktiven, individuellen Rezeption von Videos am Computer stehen eher passive Szenarien der Co-Präsenz in kleinen und großen Gruppen, beispielsweise bei öffentlichen Filmvorführungen in Kinos, gegenüber. Soziale Nutzungskontexte in der virtuellen Welt des WWW sind sehr vielseitig. Insbesondere YouTube in seiner Eigenschaft als soziales Netzwerk und *video sharing* Plattform prägte viele soziale Interaktionsformen im Zusammenhang mit Videos [Benevenuto et al. 2008; Hündgen und Gugel 2013; Snelson 2011].

Organisatorische Kontexte von *Human Video Interfaces* betreffen beispielsweise formelle Lernangebote, etwa als Teil eines LMS oder Blended-Learning-Szenarios einer Institution, aber auch Gestaltungsvorgaben (z.B. *Style Guides*), Kommunikationsrichtlinien (z.B. *Social Media Policies*).

Um die Schnittstelle zwischen Mensch und Video in einer Videolernumgebung gestalten zu können, gilt es zunächst mit den vorhandenen Gestaltungsraum vertraut werden. Bislang war nur von abstrakten Steuerungselementen, interaktiven Bedienelementen, Rückmeldungen vom System und dem Videoplayer die Rede. Um eine bessere Vorstellung vom zu gestaltenden Gegenstand zu bekommen, werden die Elemente des Gestaltungsraums im nächsten Teilabschnitt hinsichtlich ihrer räumlichen Beziehung zum Video betrachtet.

### 3.2.1 Gestaltungsraum von Hypervideos

Ein Gestaltungsraum (engl. *design space*) umfasst eine Menge möglicher Gestaltungsformen und Parameter, die man einem Produkt verleihen kann. Auf Hypervideos bezogen besteht der Gestaltungsraum zunächst aus der Fläche des Video-Frames und der Sequenz bewegter Bilder. Die räumliche und zeitliche Anordnung von Objekten innerhalb eines Gestaltungsraums bezeichnet man als Layout.

Das Video selbst unterliegt einem eigenen zeitlichen Layout, indem sich die Bilder innerhalb des Frames im Laufe der Zeit verändern respektive „bewegen“. Die Position des Frames im Raum bleibt dabei konstant, während sich über die Zeit hinweg die dargestellten Inhalte verändern. Die eigentliche Gestaltung des Videos ist jedoch nicht Teil des hier betrachteten Gestaltungsraums, da bei der Bearbeitung und Erstellung eines Hypervideos die immanente Datenrepräsentation der Medienelemente (i.d.R. Dateien) unberührt bleibt. Bedeutsam sind hingegen die Beziehungen zwischen dem Video und den übrigen Objekten. Man unterscheidet Objekte, deren Darstellung von der Medienzeit des Videos abhängt (z.B. ausgehende Links, Annotationen, synchrone Folien) von solchen Objekten, die sich auf einen Zeitpunkt im Video beziehen (z.B. Inhaltsverzeichnis, eingehende Links, aber auch Metadaten wie Titel und Autor).

Die zentrale Herausforderung bei der Gestaltung der graphischen Benutzerschnittstelle ist die bestmögliche Ausnutzung der zur Verfügung stehenden Fläche in Abhängigkeit der Zeit. Die räumliche Sichtweise auf die Anordnung der sichtbaren Objekte eröffnet eine einfache Beschreibung des Gestaltungsraums. In den nachfolgenden zwei Absätzen werden Objekte anhand ihrer Platzierung innerhalb oder außerhalb des Video-Frames beschrieben. Im Anschluss daran finden Hyperlinks, als die zentralen Strukturelemente, eine gesonderte Darstellung. Auch Zeitleisten, als räumliche Repräsentanten des zeitlichen Layouts, werden in einem gesonderten Teilabschnitt thematisiert.

**Objekte im Video** Darstellungsobjekte im Video umfassen z.B. Hyperlinks [Girgensohn et al. 2004; Sawhney et al. 1996], Untertitel, Notizen und Hervorhebungen, aber auch Kommentare und Markierungen [Vohle 2010]. Objekte dieser Kategorie besitzen weder eine statische räumliche, noch zeitliche Position, d.h. sie erscheinen weder permanent, noch immer an der selben Stelle. Die jeweiligen Objekte werden dabei in eine Schicht über dem Video eingefügt, ohne das Ausgangsmedium zu verändern [Schultes et al. 2010]. Auch im Vollbildmodus bleiben sie sichtbar. Die Objekte weisen zumeist einen inhaltlichen Bezug zur zeitlich verankerten Videoszene auf. Es erscheint deshalb naheliegend, selbige aufgrund ihres semantischen Bezugs auch an der betreffenden Stelle innerhalb des Video-Frames zu annotieren [Liestøl 1994]. Ist dies

nicht der Fall kommt das *Gestaltgesetz der Nähe* [Khzaeli 2005, S. 26-27] nicht zum Tragen und mindert daher die semantische Ausdrucksfähigkeit der Objekte.

Die Repräsentation der Objekte hebt sich deutlich von der im Hintergrund laufenden Filmszene ab. Darüber hinaus besitzen Interaktionselemente wie Hyperlinks (*spatio temporal hyperlinks* [Sawhney et al. 1996]) eine gewisse *affordance* [Norman 2002], die sie als Interaktionselemente kennzeichnet. Die zeitliche Dynamik durch die Einblendung der Objekte sowie deren auffällige Darstellung soll die Aufmerksamkeit des Betrachters erregen. Gleichzeitig kommt es aber zu einer Beeinträchtigung der Szene. Die Visualisierung der Objekte ist deshalb ein Kompromiss zwischen Erkennbarkeit und Selbstbeschreibungsfähigkeit auf der einen Seite und einer minimalen Beeinflussung der Filmästhetik auf der anderen Seite.

**Objekte außerhalb des Videos** Objekte außerhalb des Video-Frames besitzen nicht zwingend einen inhaltlichen Bezug zum Video. Sie können sich jedoch auf das gesamte Video beziehen oder sogar in Abhängigkeit vom zeitlichen Verlauf der Wiedergabe stehen. Im räumlichen Layout unterscheidet man daher *statische* von *zeitabhängigen* und *zeitbezogenen* Objekten außerhalb des Video-Frames.

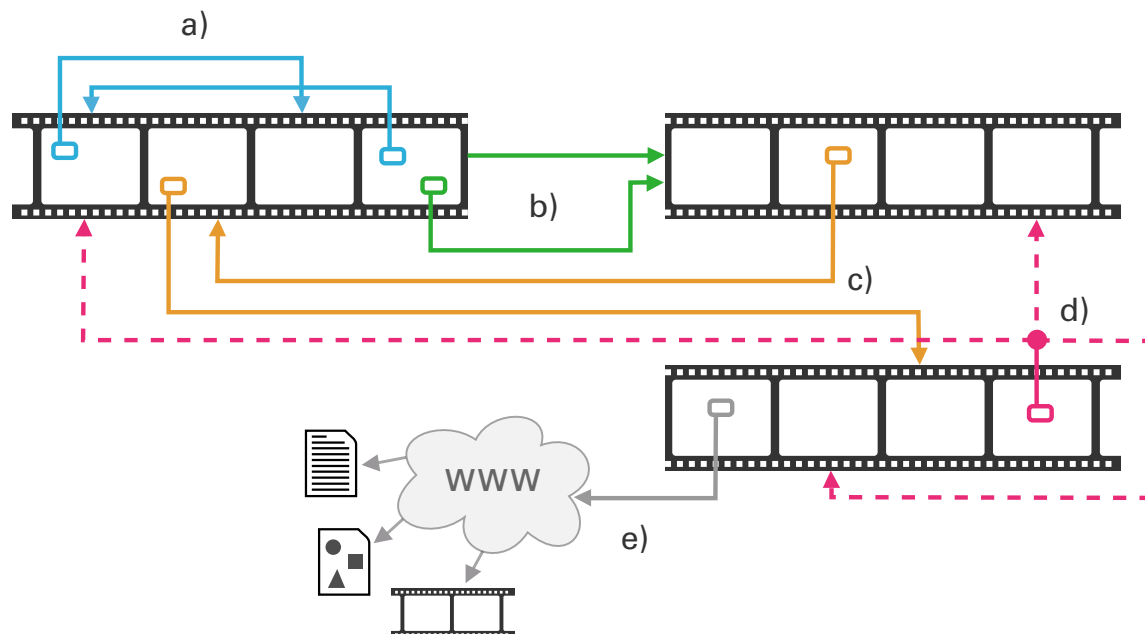
*Statische Objekte* sind unabhängig von der Abspielposition des Videos und betreffen zumeist das Video als Ganzes oder beinhalten sonstige Artefakte (z.B. die Seitennavigation, Link zum Impressum). Wird das Video im Vollbild-Modus gezeigt, werden diese zumeist ausgeblendet. Beispielhaft dafür sind Metadaten wie Titel, Autor, Datum der Veröffentlichung, Schlüsselworte und Kategorien. Aber auch semantische und personalisierte Bezüge zu anderen Ressourcen, wie z.B. ähnliche Videos oder Playlisten, sind statisch. Navigationshilfen wie zeitbezogenen Inhaltsverzeichnisse, Suchfunktionen oder *tag clouds* müssen sogar sichtbar sein, um ihren Zweck erfüllen zu können. Typisch für videobasierte Social Media Anwendungen sind die, zumeist unterhalb des Videobildes verfügbar gemachten, Verweise auf *news feeds*, Soziale Netzwerke (z.B. *facebook likes*) sowie die Anzeige von so genannten *embed codes*, um ein Video per HTML `<iframe>` in andere Web-Ressourcen integrieren zu können.

*Zeitabhängige Objekte* werden parallel zum Video dargeboten und nach dem Master-Slave-Prinzip durch die aktuelle Abspielposition des Videos gesteuert. Das Spektrum an Ausprägungen reicht von interaktiven Karten bis hin zu Webseiten und der Präsentation von Vortragsfolien [Groß et al. 2010; Ketterl et al. 2009]. In Bezug zum tragenden Video fungieren die synchronisierten Medien als visuelle Anker – sie dienen somit der Orientierung innerhalb der Medienelementzeit.

*Zeitbezogene Objekte* sind Interaktionselemente, die neben ihrer Position im zweidimensionalen Raum eine Referenz auf eine zeitliche Position in einem Zielmedium besitzen. In der Regel handelt es sich dabei um Hyperlinks, z.B. in Gestalt eines Inhaltsverzeichnisses oder einer Liste oder Ballung von (zeitbezogenen) Schlüsselwörtern, den so genannten *tags*. Aber auch Kommentare oder Diskussionsbeiträge können zeitlich verankert sein [Trahasch 2006].

**Hyperlinks** Neben der räumlichen und zeitlichen Position und Dimension sind Hyperlinks durch ihr Verweisziel sowie die sich aus der Linkaktivierung ergebenden Konsequenzen für das Quellvideo charakterisiert. Hinsichtlich des Verweisziels lassen sich innerhalb von Multimediapräsentationen drei grundlegende Formen unidirektionaler Hyperlinks unterscheiden: *intra*

*document links*, *inter document links* und *intra node links* [Hardman et al. 1999; Hofmann und Hollender 2007].



**Abbildung 3.3:** Unterscheidung von Hyperlinks anhand ihrer Verweisziele: a) *intra document links*, b) *inter document links*, c) *intra node links*, d) multidirektionale Links, e) externe Links [Seidel 2008]

*Intra document links* ermöglichen die einfache Navigation auf der Zeitleiste innerhalb einer linearen Multimediapräsentation (Abb. 3.3 / blaue Pfeile). Bei Links eines Inhaltsverzeichnisses handelt es sich beispielsweise um diese Form von Link. Die zweite Form, *inter document links*, erlaubt die Navigation in einem Netz von linearen Multimediapräsentationen (Abb. 3.3 / grüne Pfeile), wobei jeweils der Beginn eines Videos adressiert wird. Die dritte Form, *intra document links*, stellt eine Kombination der ersten und zweiten Variante dar und umfasst die Navigation innerhalb und zwischen linearen Multimediapräsentationen (Abb. 3.3 / gelbe Pfeile). Neben diesen drei Linkformen sind in der *XML Linking Language* (XLink) des W3C „extended links“ spezifiziert, die auf mehrere Ziele verweisen (multidirektionale Links) [DeRose et al. 2001] (Abb. 3.3 / rosa Pfeile). Eine fünfte Form von Links referenziert Ressourcen außerhalb der Hypervideostruktur (externe Links, Abb. 3.3 / graue Pfeile).

Wird ein Hyperlink in einem Video aktiviert, der auf ein anderes Video verweist, stellt sich die Frage, in welcher Weise die beiden Videos fortan präsentiert werden. „Die Darbietung des Zielvideos nach Aktivierung des Links kann, ebenso wie die Vorschau, hinsichtlich der spatialen Position, Dimension und Orientierung innerhalb des darstellbaren Bereichs variieren. Demzufolge kann das neue Video im Quellvideo eingebettet sein oder es vollkommen überdecken.“ [Seidel 2008]. Es muss in jedem Fall sichergestellt sein, dass die Konsequenzen einer Handlung mit den zuvor gehegten Erwartungen übereinstimmen. Dies kann durch eine eindeutige Darstellung des Hyperlinks erfolgen, aus der die genannten Szenarien hervorgehen. Dabei ist es jedoch sinnvoll, sich auf ein bis zwei oder maximal drei Szenarien zu beschränken. Eine weiterführende Analyse von zeitabhängigen Hyperlinks findet sich bei Seidel [2008, S. 28ff].

Ähnlich wie die von Bernstein [1998] aus der Erzählliteratur abgeleiteten Hypertext-Patterns, lassen sich aus den hier beschriebenen Linkformen und Konsequenzen der Linkaktivierung eine Reihe von Hypervideo-Link-Patterns ableiten und durch Beispiele belegen:

- **Detail on demand:** Je Szene im Quellvideo ist jeweils nur ein Link verfügbar, der auf ein Video mit Zusatzmaterial verweist. Nach Ablauf des Zielvideos setzt sich die Wiedergabe in der betreffenden Szene im Quellvideo fort [Girgensohn et al. 2004] (vgl. „cycle“ in Bernstein [1998]). Beispiel: *Hyper-Hitchcock* [Shipman et al. 2003a].
- **Montage:** Mindestens zwei Videoclips mit sich ergänzenden, auf einander aufbauenden Inhalten werden simultan dargestellt (vgl. „montage“ in [Bernstein 1998]). Beispiel: Multi-Angle Videos [Lu 2013].
- **Verzweigung:** Baumartige Verzweigung verknüpfen mehrere, nicht miteinander verlinkte Videoszenen oder Episoden miteinander (vgl. „sieve“ in Bernstein [1998]). Beispiel: *SIVA suite* am Beispiel der Besichtigungstour einer Wohnung [Meixner 2014, S. 93-94].
- **Kontrapunkt:** Zwei Videos sind durch Links miteinander eng verwoben, sodass die Inhalte ähnlich einem kausalen Dialog aus Rede und Gegenrede strukturiert sind (vgl. „counterpoint“ in [Bernstein 1998]). Beispiel: Film „Mörderische Entscheidung“ von Oliver Hirschbiegel<sup>8</sup>
- **Ring:** Am Ende eines jeden Videos werden mehrere Verweise auf andere Videos angeboten, sodass eine in sich geschlossene Verkettung von Videos entsteht. Beispiel: *XIMPLE* [Hurdeman et al. 2008].

**Zeitleiste** Eine Zeitleiste ist eine Metapher, der die Annahme zu Grunde liegt, die Zeit linear und progressiv abzubilden [Lindley et al. 2013, S. 3212]. Diese, häufig als waagerechter Balken dargestellte Skala gibt mit einem Zeiger Aufschluss über die aktuelle Abspielposition und dient der kontinuierlichen Navigation innerhalb der Wiedergabezeit, d.h. dem Vor und Zurückspringen innerhalb der Medienelementzeit. Funktional stellt eine Zeitleiste damit das Gegenstück zur Indexierungen eines Videos (z.B. in einem Inhaltsverzeichnis mit Kapitelsprungmarken) dar, indem ein kontinuierlicher, statt ein diskreter Zugriff auf zeitliche Positionen ermöglicht wird [Chambel und Guimarães 2002]. Auch das zeitliche Layout von Ereignissen innerhalb der Wiedergabezeit kann auf einer Zeitleiste abgebildet sein. Dies verschafft dem Betrachter einen Überblick über das Interaktions- und Kommunikationsaufkommen und gestattet einen gezielten Zugang. Auch das Nutzungsverhalten sowie die Intensität sozialer Interaktion innerhalb einer Lerngruppe bzw. Lernressource kann in Abhängigkeit der Zeit dargestellt sein [Ketterl et al. 2008; Sack und Waitelonis 2008; Seidel 2014a].

Bemerkenswert ist dabei, dass Zeitleisten von Videoplayern im Web den zeitlichen Verlauf von links nach rechts abbilden. Dies gilt auch dann, wenn die kulturspezifische Leserichtungen in umgekehrter Richtung verläuft (z.B. in arabischen Ländern).

<sup>8</sup>Zwei Versionen eines TV-Krimis wurden 1991 gleichzeitig bei zwei Fernsehsendern ausgestrahlt, sodass die Zuschauer zwischen zwei Erzählperspektiven der selben Geschichte wählen konnten [Weiberg 1994].

### 3.2.2 Usability-Herausforderungen von Human Video Interfaces

Usability als Grad der Bedienbarkeit drückt aus, in welchem Maße eine Anwendung effektiv, effizient und zur Zufriedenheit der Anwender die Bewältigung bestimmter Aufgaben ermöglicht.

In diesem Teilabschnitt finden zentrale Herausforderungen an die Usability des Human Video Interface eine Erläuterung. Grundlegend für die Betrachtung der Usability von Hypervideo-Systemen sind die ISO 14915-110:2006 [Deutsches Institut für Normung (Hsg.) 2006] sowie die ISO 9241 [Deutsches Institut für Normung (Hsg.) 2003]. In der Literatur finden sich bislang nur sehr wenige methodische Ansätze und empirische Belege für die benutzerfreundliche Gestaltung von Videolernumgebungen [Chorianopoulos und Giannakos 2013]. Meixner et al. [2011] haben Guidelines für die Evaluation interaktiver Videos aus der Literatur abgeleitet. Diese sind jedoch sehr stark an Hypertext-Systeme im WWW angelehnt und vernachlässigen die den kontinuierlichen Medien innewohnenden Eigenschaften (z.B. zeitabhängige Darstellung).

**Zeitliche Dichte an Ereignissen** Ein Hypervideo enthält mehr Informationen als das zugrunde liegende Video. Aus diesen Zusatzinformationen können sich Schwierigkeiten ergeben, die einerseits mit dem zeitlichen Ereignis selbst und andererseits mit der Anzahl an Ereignissen verbunden sind [Chambel und Guimarães 2002]. Wenn zu viele Objekte gleichzeitig zu sehen sind und sich gegenseitig überlagern, kann das den Betrachter beim Versuch eine Auswahl relevanter Objekte zu treffen überfordern (vgl. *intrinsic cognitive load*, S. 36). Die Anzahl an synchronen Annotationen sollte daher zusammen mit den im Video enthaltenen Symbolen begrenzt sein. Je weniger Annotationen, desto besser, denn schon ein einziges Ereignis, was mit der Videowiedergabe einhergeht und diese visuell oder akustisch interferiert, kann eine Ablenkung bzw. Unterbrechung des Rezipienten bewirken [Hoffmann und Herczeg 2006]. Die von [Miller 1956] empfohlene *magical number*  $7 \pm 2$  kann hier als Richtwert gelten.

Wie oben bereits erwähnt, eignen sich Zeitleisten dazu, die zeitliche Folge an Annotationen im zweidimensionalen Raum zu visualisieren. Ohne weiterführende Visualisierungsansätze kann diese Darstellungsform in Anbetracht der Häufigkeit und Form von Annotationen nicht skalieren. In Betracht kommen beispielsweise mehrere Spuren (engl. *tracks*), Filter nach Art des Ereignisses oder das Heranzoomen von Clustern, welche eine zeitliche Ballung verschiedener Ereignisse repräsentieren.

**Anzeigedauer** Damit der Anwender eine Chance hat annotierte Inhalte zu verarbeiten oder ein Bedienelement auszulösen, stellt sich die Frage, wie lange eine Annotation oder etwa ein Link mindestens sichtbar bleiben muss. Will man für diese Objekte eine bestimmte Zeitdauer als untere Schranke der Anzeige definieren, so muss dies auch in Abhängigkeit der Größe des Linkankers und in Relation zu den Abmessungen des Bildschirms erfolgen [Sawhney et al. 1996]. Ungeachtet dieses in *Fitts's Law* [Fitts 1954] zum Ausdruck kommenden Zusammenhangs, kann eine solche Festlegung nicht verhindern, dass die Anzeige eines Hyperlinks als zu kurz empfunden wird. Das menschliche Zeitgefühl hängt mit dem Maß an geistiger Aufmerksamkeit zusammen. Eine als langweilig empfundene Filmszene wird demnach als länger andauernd wahrgenommen, als eine spannendere, selbst wenn beide objektiv gleich lang sind. Das gilt analog für Annotationen, die während solcher Szenen auftreten. Im Hinblick auf die Barrierefreiheit ist es deshalb ratsam, die minimale Anzeigedauer eines Objektes durch den



Anwender in der Wiedergabekomponente festlegen zu lassen [Seidel 2008]. Alternativ dazu kann sich die Anzeigedauer z.B. durch das Gedrückthalten einer Taste oder die Annäherung des Mauszeigers verlängern. Zeitleisten erlauben darüber hinaus Rückschlüsse über die verbleibende Zeit der Linkanzeige.

**Lost in video hyperspace** Innerhalb des zeitlichen Layouts eines Videos sowie innerhalb umfangreicher Hypervideo-Netze bedarf es Orientierungshilfen für Anwender [Debevc et al. 2008, S. 33]. Im Hinblick auf die Selbstbeschreibungsfähigkeit eines Hypervideo-Systems ist die aktuelle Position im *hyperspace* und innerhalb der Abspielzeit des Videos kenntlich zu machen.

Wenngleich nicht immer die gesamte Struktur der Hyperlinks darstellbar ist, sollten zumindest die angebotenen Verknüpfungen, d.h. die nächsten Nachbarn im Netz, soweit ersichtlich und aussagekräftig dargestellt sein, dass der Anwender gemäß seiner Erwartungen navigieren kann. Folgende Fragen sollte der Anwender für sich beantworten können [Chambel und Guimarães 2002]:

- Wo befindet sich der Link im Video?
- Welches Ziel hat der Link?
- Welche Szene / welchen Bereich adressiert der Link im Zielmedium?
- Ab wann steht der Link zur Verfügung?
- Wie lange bleibt ein Bedienelement sichtbar und benutzbar?
- Wie gelangt man nach der Aktivierung eines Hyperlinks wieder zurück an dessen Ausgangspunkt?

Eine weitere software-ergonomische Anforderung betrifft die Steuerbarkeit des Systems innerhalb eines Hypervideo-Netzes. Während die meisten Hypertext-Systeme (z.B. Web-Browser) die Historie der Seitenaufrufe speichern und somit ein Vor- und Zurückspringen entlang des Navigationspfades ermöglichen, ist dies auch für zeitgenaue Sprünge zwischen verknüpften Videos erforderlich.

Zusammengefasst stehen die Forderungen nach Selbstbeschreibungsfähigkeit, Erwartungskonformität und Steuerbarkeit im Einklang mit den Grundsätzen der Dialoggestaltung (EN ISO 9241-110:2003 „Ergonomie der Mensch-System-Interaktion“ [Deutsches Institut für Normung (Hsg.) 2003]).

**Barrierefreiheit** Die Herausforderung einer barrierefreien Gestaltung der Schnittstelle zwischen Anwender und Video resultiert maßgeblich aus der kritischen Zugänglichkeit des Mediums Video selbst. Schwierigkeiten ergeben sich damit für Menschen, die physischen oder psychischen Einschränkungen unterliegen. Die Art und Weise, in der Menschen bei der Benutzung technischer Systeme eine Benachteiligung erfahren können, ist dabei sehr unterschiedlich. Betroffen davon sind diverse Personengruppen in unterschiedlichen Lebenssituationen. Im Kontext des E-Learnings betrifft dies nicht nur Lernende mit verminderter Seh- oder Hörleistung, sondern auch Menschen mit einer langsamen Reaktionsgeschwindigkeit oder verminderten Sprachkenntnissen. Zudem stellt die Gewährleistung einer barrierefreien Nutzung von Systemen zur Informationsverarbeitung mit akustischen und visuellen Informationsquellen (z.B. Videolernumgebungen) auch eine gesetzliche Pflicht dar (vgl. BGB, Gesetz zur Gleichstellung

behinderter Menschen, Anlage (zu den §§ 3 und 4 Abs. 1). Bestimmte Gestaltungselemente interaktiver Videos sowie alternative Medienrepräsentationen können Anwendern mit physischen oder psychischen Einschränkungen jedoch die Betrachtung und Bedienung erleichtern. Exemplarisch sei an dieser Stelle auf Untertitel und Transkriptionen der Audioinhalte sowie die Vergrößerung von Bildelementen und die Verlangsamung der Wiedergabegeschwindigkeit verwiesen.

Obwohl es keine speziellen Richtlinien für Videoanwendungen gibt, finden sich in den W3C-Empfehlungen *WAI-ARIA* [W3C 2011c] und *WCAG2* [Caldwell et al. 2008] zielführende Hinweise für die mediale sowie software-technische Umsetzung. Um die Anforderungen verschiedener Gruppen in unterschiedlichen Situationen zu erfüllen, sind in den *WCAG2* drei Konformitätsstufen („level“) definiert: A (niedrig), AA und AAA (hoch). Je höher die Stufe, desto größer ist der zu betreibende Aufwand, um beispielsweise visuelle Informationen durch adäquate Audiobeschreibungen zu ergänzen.

Um die Usability von Systemen besser einschätzen und miteinander vergleichen zu können, wurden diese Herausforderungen in eine erweiterte Fassung der Guidelines von Meixner et al. [2011] integriert (vgl. Anhang A.2, S. 261). Die Guidelines dienen sowohl zur Konzeption und Entwicklung gebrauchstauglicher Anwendungen (vgl. S. 105), als auch als Bewertungsinstrument für bestehende Hypervideosysteme. Im Hinblick auf die Beschreibung von wiederkehrenden Gestaltungslösungen in Form von Design Patterns, markieren sie auftretende Wechselwirkungen eines Problems bzw. Konsequenzen einer umgesetzten Lösung.

## Technische Herausforderungen für den Einsatz von Hypervideos im WWW

Seit der Umsetzung von HTML5 und CSS3 in modernen Web-Browsern können Videos nativ, d.h. ohne externe Programmkomponenten wiedergegeben werden. Die gängigen Browser unterstützen jedoch unterschiedliche Video-Codecs, ohne dass man einen Codec als kleinsten gemeinsamen Nenner für alle Browser und Plattformen empfehlen könnte (vgl. Anhang A.3, S. 267). Auch in der Implementierung der Video-Unterstützung gibt es Unterschiede, die bei der Entwicklung zu berücksichtigen sind.

Browser-Plugins wie *Adobe Flash* bzw. *Flex*, *Microsoft Silverlight* oder *JavaFX* (via *Java Webstart* oder *Java Applets*) könnte man daher als eine Alternative ansehen, da sich diese Anwendungen auf unterschiedlichen Systemen gleich verhalten und darstellen lassen. Die verarbeiteten Datenformate entziehen sich jedoch den offenen Web-Standards und sind darüber hinaus mit erheblichen Nachteilen verbunden, welche ihre Vorteile überwiegen:

- Die Plugins bzw. Applets sind auf einigen Endgeräten nicht lauffähig (z.B. laufen *Adobe Flash* und *MS Silverlight* nicht auf den Apple-Geräten *iPhone* und *iPad*).
- Plugins führen teilweise zu einer hohen CPU-Last durch den Browser.
- *Java Applets* weisen eine lange Ladezeit auf.
- Die Nachhaltigkeit ist fraglich, da die Plugins, im Gegensatz zu den auf offenen Standards basierenden Browser-Technologien, in den meisten Fällen von einem einzigen Anbieter bereitgestellt werden und somit einer Produkt- und Vermarktungsstrategie unterliegen.



- Anbieterunabhängige Plugin-Implementierungen wie der Open Source SWF-Player *gnuash* stellen eine unzureichende Alternative dar, da sie ältere Versionen von SWF-Dateien nicht unterstützen.

Eine Implementierung auf Basis internationaler, herstellerunabhängiger Standards bzw. den vom W3C empfohlenen technischen Spezifikationen und Richtlinien ist somit eine Grundanforderung an die Entwicklung eines Hypervideo-Players und -Frameworks. Folgende Spezifikationen erfüllen diese Anforderung: HTML5 [Berjon et al. 2012], CSS3 [W3C 2011a] und *ECMAScript* [W3C 2011b], auch bekannt als JavaScript. Die Web-Standards und Empfehlungen sind auf einen breiten Konsens und weitreichende Nutzbarkeit ausgerichtet. Für interaktive Videos existieren bislang noch keine Standards. In Bezug auf die Implementierung von interaktiven Videoplayern und zeitbasierten Medienpräsentationen ergeben sich jedoch einige technische Herausforderungen, welche in den folgenden Teilabschnitten beleuchtet werden.

### 3.3.1 Download und Cache-Management / Echte Nicht-Linearität

Durch die Einblendung von Zusatzinformationen und die nicht-lineare Organisation der Videoinhalte kann das Datenaufkommen bei der Wiedergabe und der Navigation im Vergleich zu gewöhnlichen, linearen Videos erheblich zunehmen. Zusatzinformationen können aus Texten, Bildern, weiteren Videos oder Webseiten bestehen. Die Navigation in einer nicht-linearen Struktur mehrerer Videos ist durch den Wechsel von Videos und der Wiedergabe von Fragmenten gekennzeichnet. Benutzer folgen dabei durchaus unterschiedlichen Pfaden durch die Struktur der miteinander verlinkten Videos. Dabei sind Verzögerungen bei der Datenübertragung nicht ausgeschlossen. Sie äußern sich in Unterbrechung der Wiedergabe und Störung der filmischen Ästhetik („fluid viewing experience“ [Meixner und Hoffmann 2012]). Die Herausforderung besteht darin, die Verzögerungen so weit zu minimieren, dass sie der Anwender nicht bemerkt („real’ non-linearity“). Übliche Realisierungen der Nicht-Linearität beruhen auf Verlinkung oder Zeitverzögerung. Die Verlinkung besteht zwischen Webseiten, die beim Aufruf eines Links neu laden und somit einen medialen Bruch darstellen (z.B. Links in *YouTube*). Bei einer Realisierung durch Zeitverzögerungen werden lediglich einzelne Elemente einer Webseite (z.B. das Video oder Videoszenen bei Sprüngen auf der Zeitachse) ausgetauscht, deren Download und Darstellung dennoch Zeit beansprucht. Die benötigte Zeit für den Download variiert je nach Datenübertragungsrate und -volumen sowie dem Cache-Management, wodurch gewisse Inhalte im Zwischenspeicher (engl. *cache*) des Webbrowsers behalten oder gelöscht werden. Die Caching-Strategien der Web-Browser sind durchaus unterschiedlich. Während Browser auf Desktop-Rechnern unter Umständen das gesamte Video in den Cache laden, puffern mobile Endgeräte weniger Daten.

Auch im Umgang mit Annotationen besteht die Herausforderung darin, nur die zur Wiedergabe benötigten Elemente zu laden und mehrfache Downloads durch Caching-Strategien zu vermeiden. Dabei können die Hypervideo-Strukturen berücksichtigt werden, indem Medienelemente bereits im Voraus, etwa mit Hilfe von Markov-Ketten [Grigoras et al. 2002], geladen werden (*prefetching*) [Meixner und Hoffmann 2012]. Die Zwischenspeicherung kann, anstatt im Browser-Cache, auch lokal in einer *Indexed Database* [Mehta et al. 2013] des Browsers erfolgen.

### 3.3.2 Synchronisierung von Video und Annotationen

Zeitsteuerung und Synchronisierung von Medienelementen ist im Gegensatz zu *SMIL* nicht Teil des HTML5-Standards. Zur Umsetzung von zeitabhängigen Ereignissen (z.B. Einblendung eines Hyperlinks) oder zeitlich getakteten Synchrondarstellungen von Video und Bild bedarf es zusätzlicher client-seitiger Implementierungen. Die Performanz von Algorithmen ist dabei umso schlechter, je höher die Anzahl an Annotationen ist und je genauer diese zeitlich synchronisiert sind. Limitierend wirkt hierbei die Performance des Web-Browsers, in einem Intervall von  $n$  Millisekunden zu prüfen, welche der  $m$  Annotationen ein- oder auszublenden sind. Ein einfacher Algorithmus (siehe Algorithmus 1) erfasst in einem definierten Zeitintervall die aktuelle Wiedergabezeit (Medienzeit) und durchläuft die Liste der Annotationen. Dabei wird geprüft, ob die Medienzeit innerhalb von Anzeigebeginn und -dauer der Annotation liegt. Falls ja, erfolgt die Darstellung der Annotation („show()“), ansonsten wird sie ausgeblendet („hide()“). Der Aufwand für die Suche nach aktuellen Annotationen je Zeitintervall ist dabei linear und somit verbesserungswürdig.

**Input :**

$A$  Liste der Annotationen eines Videos

$t$  aktuelle Abspielposition des Videos

```

Function displayAnnotation()
  for  $a \in A$  do
    if  $a.start \leq t \leq a.end$  then
      |  $show(a)$ 
    else
      |  $hide(a)$ 
    end
  end
  return
End

```

**Algorithmus 1 :** Ein- und Ausblendung von Annotationen in Abhängigkeit der Wiedergabezeit je Zeitintervall der Prüfung

### 3.3.3 Adressierung und Abruf von Medienfragmenten

Im Vergleich zu Bildern oder Texten lassen sich Video- und Audio-Elemente im WWW oftmals nur als Ganzes abrufen und adressieren. Der direkte Zugriff sowie die Verlinkung dieser Medien ist damit nur eingeschränkt möglich [Mannens et al. 2012]. In manchen Anwendungsszenarien von Hypervideos ist es sinnvoll nur einen zeitlichen Ausschnitt oder nur einen bestimmten Bildbereich im Player zu übertragen. Ein Beispiel für zeitliche Fragmente findet sich bei *Metavid*, wo Passagen aus Redebeiträgen von Abgeordneten aus dem U.S. Senat und Abgeordnetenhaus abgerufen werden können, ohne dabei die gesamte Senatssitzung übertragen zu müssen [Dale et al. 2009]. Abgesehen von den Zeitspannen gibt es auch eine Reihe von Anwendungen für die genaue Adressierung eines Zeitpunkts innerhalb der Medienelementzeit. Zeitgenaue URIs sind vergleichbar mit Lesezeichen in einem Buch, die statt des gesamten Buches nur ei-

ne Seite kennzeichnen. Ein breites Anwendungsfeld stellen zeitbezogene Links dar, wie sie bei YouTube innerhalb von Kommentaren auf eine Stelle im Video verweisen oder in Hypervideos vorkommen. Die Adressierung von räumlichen Fragmenten („region of interest“ / ROI) ist u.a. bei hochauflösenden Videobildern oder Panoramaaufnahmen sinnvoll. Ein Beispiel hierfür liefert die Anwendung *Diver*<sup>TM</sup> [Pea 2006], mit deren Hilfe Lernende Ausschnitte eines 360°-Videos kommentieren und diskutieren können.

Um Medienfragmente sowohl zeitlich, als auch räumlich über das HTTP-Protokoll als *Uniform Resource Identifiers* (URI) adressieren zu können, hat eine Arbeitsgruppe beim W3C eine Empfehlung für *Media Fragments URI* erarbeitet [Hausenblas et al. 2011; Mannens et al. 2012]. Diese Empfehlung ist unabhängig von Medienformaten und bezieht sich auf zeitlich lineare Medien inklusive der enthaltenen Medienspuren (z.B. Video, Audio, Untertitel, etc.). Bezüglich des Adressschemas für den zeitlichen Versatz wird zwischen *fragments* („#“) und *queries* („?“) unterschieden. Mit Hilfe von Fragments stellt der Client (z.B. Webbrowser) nur einen Teil der Ressource zur Verfügung. Sofern ein Web-Server *Byte range requests*<sup>9</sup> unterstützt, muss auch nur das angefragte Fragment heruntergeladen werden. Mittels *queries* generiert der Web-Server aus dem adressierten Medium eine neue Ressource und liefert sie aus. Zur Angabe eines zeitlichen Versatzes und der Dauer des Medienfragments von on-demand Medien sind drei verschiedene Schemata vorgesehen. Darüber hinaus lassen sich einzelne Medienspuren (z.B. die englische oder französische Audiospur) und benannte Segmente („named dimension“), wie z.B. Kapitel, ansprechen<sup>10</sup>. In der nachfolgenden Auflistung 3.1 sind einige Beispiele zu sehen.

```

1 # Zeit-Schemata
2 t=npt:<start-in-seconds>,<end-in-seconds>
3 t=npt:<m>,<s>.<ms>:<h>:<m>.<ms>
4 t=smpte-<frame-rate>:<h>:<m>:<s>,<h>:<m>:<s>.<ms>
5
6 # Zeit Beispiele
7 t=npt:120,0:02:01.5           # ergibt Zeitintervall von [120,121.5[
8 t=npt:10,20                  # ergibt Zeitintervall von [10,20[
9 t=,20                         # ergibt Zeitintervall von [0,20[
10 t=smpte-25:0:02:00,         # ergibt Zeitintervall von [120,end[
11
12 # Bildausschnitte
13 xywh=<einheit>:<x>:<y>:<width>:<height> # Schema
14 xywh=100,200,320,240         # 320x240 Pixel große Box bei x=100 und y=200
15 xywh=pixel:100,200,320,240   # 320x240 Pixel große Box bei x=100 und y=200
16 xywh=percent:25,25,50,50     # 50%x50% große Box bei x=25% und y=25%
17
18 # Tracks
19 track=1&track=2              # die Tracks "1" und "2" werden extrahiert
20 track=video                  # nur der Track "video" wird extrahiert

```

<sup>9</sup>Der Client sendet einen *HTTP Range request header*, der vom Server mit einer *HTTP 206 Partial Content response* erwidert wird.

<sup>10</sup>Die Adressierung von *tracks* erfordert eine serverseitige Auswertung der Meta- bzw. Header-Daten. Nur wenige VideoCodes enthalten diese Informationen. Die Zeitangaben zu benannten Abschnitten sind i.d.R. nicht im Header enthalten, sie können jedoch durch Annotationen ergänzt werden.

21	track=Lecture-Video	# nur der Track "Lecture-Video" wird extrahiert
22		
23	# Benannte Segmente	
24	id=1	# Segment "1" wird extrahiert
25	id=Kapitel-1	# Segment namens "Kapitel-1" wird extrahiert
26	id=Ausblick	# Segment namens "Ausblick" wird extrahiert

**Auflistung 3.1:** Schemata und Beispiele der *Media Fragment URI* für die Zeit, Bildausschnitte und Tracks sowie benannte Segmente [Hausenblas et al. 2011; Mannens et al. 2012]

Diese Empfehlungen wurden bislang zumindest in den verbreiteten Webservern *Apache 2* und *nginx* hinsichtlich der Zeitfragmente berücksichtigt, während jedoch nur sehr wenige Videoplayer eine Schnittstelle dafür bereitstellen. *YouTube* begegnet diesen Standardisierungs-bemühungen mit einem eigenen URI-Schema, was sich aus der ID des Videos und den anein- andergereihten Zeitangaben in Minuten und Sekunden zusammensetzt:

1	http://www.youtube.com/watch?v=<video-id>#t=<m>m<s>s
---	--

**Auflistung 3.2:** YouTube URI-Schema für Medienfragmente unter Angabe der eindeutigen Video-Id sowie den Minuten (<m>) und Sekunden (<s>) der Abspielposition

#### 3.3.4 Deklarative Ansätze der Repräsentation von Multimedia

Wie bereits in Abschnitt 3.1.2 (S. 61) erwähnt, besitzen Multimedia-Dokumente gegenüber gewöhnlichen Web-Dokumenten eine zeitliche Dimension. Diese Dimension kann aus einem kontinuierlichen Medium hervorgehen, aber auch aus zeitlich strukturierten diskreten Medien (zeitliches Layout) sowie aus einer Animation heraus entstehen, die ein Programmcode ausgeführt [Cazenave et al. 2011]. In der Umsetzung lassen sich imperative von deklarativen Ansätze unterscheiden, wobei auch Mischformen gebräuchlich sind. Der imperative Ansatz ermöglicht Benutzerinteraktionen und zeitliche Darstellungsänderungen durch die Ausführung von (Software-)Skripten. Hierbei sind generische Skripte statischen Implementierungen (z.B. *HTML* [Chambel und Guimarães 2002] und *HyperCafe* [Sawhney et al. 1996]) überlegen, da sie i.d.R. flexibel auf andere Kontexte übertragbar sind [Sadallah et al. 2012]. Deklarative Ansätze beruhen auf Auszeichnungen im Dokument, die vom Wiedergabesystem interpretiert und dargestellt werden. Dies hat insbesondere den Vorteil, dass man Dokumente in verschiedenen Anwendungen wiedergegeben kann. Die Pflege und Wiederverwendung von Dokumenten innerhalb von Workflows oder Autorenumgebungen gestaltet sich durch deklarative Ansätze leichter. Im Hinblick auf die Barrierefreiheit, sind standardkonforme Auszeichnungen eine Voraussetzung für eine semantische Anreicherung der Web-Inhalte, die Menschen mit körperlichen Beeinträchtigungen den Einsatz assistiver Technologien ermöglicht [W3C 2011c]. Web-Browser unterstützen die Wiedergabe von deklarativen Multimedia-Dokumenten inklusive ihrer zeitlichen Komponente jedoch nur ungenügend. Insbesondere die spezifizierten Multimedia-Merkmale von HTML5 und CSS3 sind dahingehend unzureichend [Berjon et al. 2012]. Es fehlen Attribute, um zeitgenaue und synchrone Ereignisse darzustellen. Im Gegensatz dazu eröffnet die *Synchronized Multimedia Integration Language* (SMIL) die Möglichkeit, Dokumente anhand der Zeit zu strukturieren. Mit Hilfe von SMIL kann man wiederum die räumliche und logische Anordnung von

Elementen nicht im selben Maße ausdrücken, wie etwa in HTML5 [Berjon et al. 2012; Bulterman et al. 2008]. Web-Browser unterstützen auch keine SMIL-Dokumente. Darüber hinaus fehlt es an Playern, die den Standard vollständig umsetzen. Die Weiterentwicklung von SMIL3 wird vom W3C nicht mehr forciert [Bulterman et al. 2008]. Ähnlich verhält es sich mit der *Timed Text Markup Language*, die für zeitabhängige Annotationen konzipiert wurde und somit weder zeitbezogene Interaktionselemente, noch weiterführende spatiale Layout-Attribute bereithält [Dolan et al. 2013].

Als XML-basierter Standard für die Annotation von Multimedia mit Metadaten stellt MPEG-7 Deskriptoren (*descriptors*) und Beschreibungsschemata (*description schemes*) bereit, mit denen man jedoch keine expliziten Beziehungen zwischen diesen Entitäten deklarieren kann [Pattanasri et al. 2007]. Zudem unterstützen derzeitige Browser kein MPEG-7.

In *Timesheet.js* werden SMIL-Deklarationen mittels JavaScript in HTML und CSS3 übersetzt und ausgegeben [Cazenave et al. 2011]. Dieser Kompromiss zwischen einer deklarativen und einer imperativen Implementierung erfüllt alle Layoutanforderungen hinsichtlich zeitlicher und räumlicher Dimensionen inklusive grundlegender, zeitabhängiger und zeitbezogener Benutzerinteraktionen. Fortgeschrittene Konzepte zyklischer und multidirektionaler Links sind mit *Timesheet.js* jedoch nicht realisierbar.

Hinsichtlich einer Integration von Hypervideo-Dokumenten in textbasierte Autorensysteme sind auch vereinfachte Deklarationen wie *Wiki Creole* oder *markdown* in Betracht zu ziehen. Anhand einer überschaubaren Menge an Ausdrücken können Autoren kontinuierliche Medien zeitlich und räumlich arrangieren. Die Anwendung übersetzt die Ausdrücke in standardkonforme Deklarationen und stellt die Medien entsprechend festgelegter Gestaltungskonventionen dar. Beispielanwendungen wie das *VI-WIKI* [Seidel 2008, 2012b] erweitern die kollaborative Textverarbeitung in Wiki-Systemen um solche Ausdrücke zur zeitlichen Strukturierung (siehe S. 246).

## Produktion und Integration in Lernumgebungen

Ein Grundvoraussetzung, um Hypervideos beispielsweise in der Lehre einsetzen zu können, besteht in der Bereitstellung von adäquaten Werkzeugen zur Produktion hypervideobasierter Lernmaterialien. Diese Notwendigkeit ist unabhängig davon, ob Dozierende, Lernende oder Dritte in die Produktionsprozesse eingebunden sind. Ziel dieses Abschnitts ist es, gemäß dem aktuellen Stand der Forschung, konkrete Anforderungen für die Entwicklung und Gestaltung von Autorenumgebungen als ein Bestandteil von Videolernumgebungen abzuleiten. Unter einer Autorenumgebung wird hier eine Anwendung verstanden, mit der vorhandene Videos angereichert sowie miteinander verknüpft und arrangiert werden können.

Im folgenden Abschnitt werden zunächst prozessorientierte Merkmale und Ausprägungen der Produktion von Hypervideos erläutert, bevor einzelne Vorgehensweisen und deren Repräsentation in Autorenprogrammen betrachtet werden. Der zweite Teil gewährt einen Überblick über integrierte Ansätze von Entwicklungsumgebungen. Die Im Fokuss steht dabei die Integration in herkömmliche Lernumgebungen mit Hilfe festgelegter Schnittstellen oder standardisierter Dateiformate.

Die Produktion von Hypervideos baut auf vorhandene Inhalte, insbesondere auf Videos auf und zielt auf eine semantische, strukturelle und inhaltliche Anreicherung der Ausgangsmedien. Eine semantische Anreicherungen der Videos eröffnet die Suche und den Abruf von zeitbasierten Informationen und erleichtert somit den Zugang zu nicht-linearen, sukzessive präsentierten Videoinhalten. Strukturelle Anreicherungen schaffen räumliche und zeitliche Beziehungen zwischen Inhalten in verschiedenen Medien. Daraus resultieren beispielsweise sequentielle, parallele und *on-demand* Darstellungen sowie baum- oder netzartige Inhaltsstrukturen [Tidwell 2005, S. 80-82]. Inhaltliche Anreicherungen besitzen neben ihrer semantischen und strukturellen Verankerung einen informationellen Mehrwert für den Rezipienten, d.h. sie ergänzen das Video um Zusatzinformationen und gegebenenfalls Lerninhalte. In der Regel umfasst die Produktion mehrere, in ihrer Reihenfolge variable Arbeitsschritte:

1. *Medienerstellung*: Erstellung und Bearbeitung der zur Darstellung bestimmten Medienobjekte [Hardman 2004];
2. *Metadaten*: Angabe von Metadaten je Medienobjekt;
3. *Zeitliches Layout*: zeitgenaue Auszeichnung durch Artefakte (z.B. Kapitelmarken, Hervorhebungen, Untertitel, Verweise) und Synchronisierung mit anderen Medien;
4. *Räumliches Layout*: Arrangement der Objekte im Darstellungsbereich [Hardman 2004];
5. *Festlegung der Interaktionen*: Verknüpfung mehrerer Videos durch zeitgenaue Verweise, wie z.B. Kapitelmarken oder Hyperlinks;
6. *Evaluation*: Test und Revision.

Darüber hinaus stellt auch die Konfiguration des Autorensystems, insbesondere die Auswahl von Funktionen bzw. Werkzeugen zur Hypervideo-Produktion, einen Arbeitsschritt dar. Je nach Werkzeug eröffnen sich Möglichkeiten für eine schrittweise kollaborative Weiterentwicklung von Hypervideos, sodass die Grenzen zwischen Produzenten und Konsumenten verwischen. In offenen Kollaborationsprozessen ist der jeweilige Zustand in einem Bearbeitungszyklus nicht vorhersehbar. Rollenfestlegung zur Moderation und effektiven Verwaltung der Nutzerbeiträgen sind wie bei anderen Anwendungen des *Social Web* notwendig. Dies hat nicht nur Konsequenzen für die Planbarkeit eines Hypervideo-Angebots, sondern auch für die Gestaltung und Usability einer Autorenumgebung (bzw. eines Players), die leicht erlernbar sein sollte.

#### **3.4.1 Produktion: Vorgehensweisen und Paradigmen**

Die oben genannten Arbeitsschritte lassen bereits erkennen, dass sich der Produktionsprozess nicht an Regeln der Filmproduktion orientiert [Lehner und Siegel 2009]. Es geht also nicht um die Herstellung von filmischen Materialien (vgl. Entwicklung von Storyboards und Filmproduktion in: Stahl et al. [2005]), sondern um eine gesonderte Form der Post-Produktion. Die dabei angewandten Vorgehensweisen finden auf einer organisatorischen und einer operativen Ebene statt. Auf der organisatorischen Ebene geht es um die am Prozess beteiligten Akteure und deren Intention des Mitarbeitens. Auf der operativen Ebene stehen mögliche Vorgehensweisen der Akteure im Blickpunkt.



Auf der Ebene der Organisation des Produktionsprozesses gibt es vier grundlegende Verfahrensweisen, die während der Umsetzung miteinander kombinieren werden können:

- **Produzent-Rezipient:** Ein einzelner Autor oder eine kleine Gruppe von Autoren produzieren ein Lernangebot als Hypervideo, deren Anwender die Inhalte weitestgehend rezipieren, anstatt weitere Elemente zu ergänzen.
- **Automatisierung:** Durch eine Analyse der Video- und Audiodaten werden semantische Auszeichnungen, Beziehungen und Zusatzinformationen automatisch generiert und in einem Hypervideo-Lernangebot integriert [Martin et al. 2005].
- **Crowdsourcing:** Der Initiator eines Lernangebots definiert die zur Verfügung stehenden Werkzeuge und Zugriffsrechte, damit eine nicht näher eingegrenzte Gruppe von Anwendern die Videos aufbereiten kann [Morisse und Sempff 2005; Sack und Waitelonis 2008]. Ob dies gemäß den Vorstellungen des Initiators geschieht, ist nicht steuerbar.
- **Aufgabe:** Ähnlich dem Crowdsourcing-Ansatz, wobei eine bestimmte Gruppe von Anwendern im Zuge der Bearbeitung einer Lernaufgabe Videos mit Annotationen, etc. anreichert [Pea und Lindgren 2008; Stahl et al. 2006; Zahn et al. 2005].

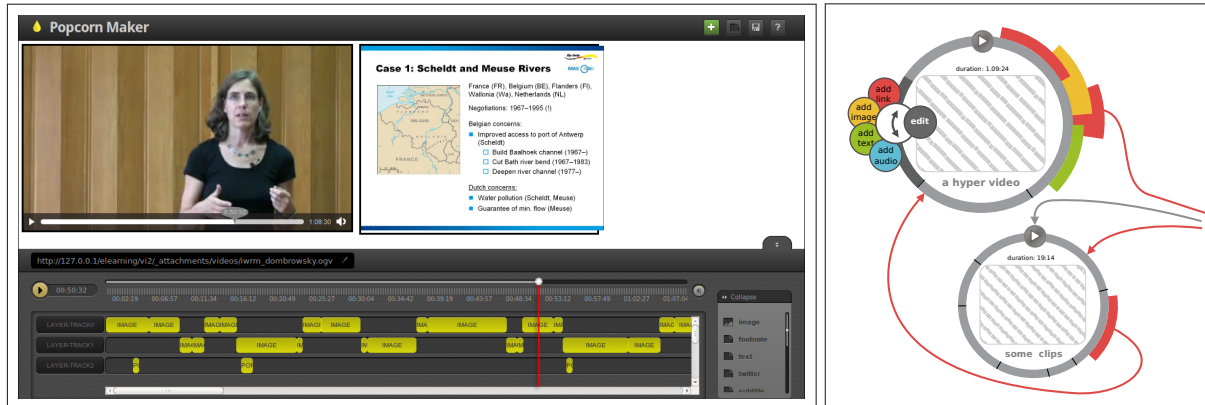
Meixner et al. [2012, S. 7ff] unterscheiden auf operativer Ebene zwei prinzipielle Vorgehensweisen: *top down* und *bottom up*. Beim *top down* Ansatz wird zunächst die Gesamtstruktur der Videokollektion entworfen, bevor man die Videos im einzelnen plant und inhaltlich präzisiert. Die grafische Repräsentation wird zuletzt gestaltet. Der *bottom up* Ansatz geht von einer Menge an Videos aus, die man unmittelbar aufbereitet und miteinander verknüpft. Im praktischen Gestaltungsprozess können diese beiden Ansätze jedoch in einander übergehen, wenn z.B. entworfene Linkstrukturen iterativ in den einzelnen Videos überarbeitet werden [Seidel 2012a].

Schlussendlich implizieren Autorenumgebungen eine bestimmte Vorgehensweise bei der Konstruktion und Gestaltung von Hypervideos. Laut Hardman [2004, S. 103] liegen diesen Anwendungen vier Paradigmen und damit auch Sichtweisen auf Multimediarepräsentationen zugrunde:

- **Zeitleiste:** Die Medienobjekte sind entlang einer oder mehrere Zeitleisten entsprechend ihrer zeitlichen Ausdehnung und Position platziert. Objektmanipulationen beziehen sich nur auf das jeweilige Objekt, nicht auf Gruppen von Objekten. Verweisziele sind auf der Zeitleiste genau definiert. Beispiel: *Mozilla Popcorn Maker* [Mozilla-Foundation 2012] (siehe Abb. 3.4, links).
- **Struktur:** Medienobjekte sind in hierarchischen oder netzartigen Strukturen und Teilstrukturen gruppiert und lassen sich einzeln und gemeinsam manipulieren. Manipulationen von Eigenschaften einer Struktur bzw. Teilstruktur wirken sich auf alle untergeordneten Objekte aus. Verweisziele werden durch Auswahl eines Medienobjekts bestimmt. Beispiel: In *Hyper-Hitchcock* sind Videos und Links hierarchisch in so genannten *Composites* organisiert [Shipman et al. 2003b]. Die Hypervideoproduktion im *SIVA Producer* basiert auf einem „Scene Graph“, indem Videoszenen direkt oder indirekt durch Auswahlknoten (Hyperlinks) miteinander verbunden sind [Meixner et al. 2010].
- **Flußdiagramm:** Ähnlich dem Konzept des visuellen Programmierens ist eine Medienrepräsentation durch eine Folge von Kommandos und Prozeduren visuell beschrieben.

Verweise erfolgen durch Angabe von Prozeduren, während jedoch zeitliche Parameter keine Berücksichtigung finden. Beispiel: *MediaLoom* [Tolva 2007].

- **Script<sup>11</sup>**: Das zeitliche und räumliche Layout einer Medienrepräsentation, einschließlich ihrer Verweise, werden durch Ausdrücke einer Programmier- oder besser Skriptsprache beschrieben. Beispiel: *Adobe Flash Professional CC*



**Abbildung 3.4:** Zwei Beispiele für zeitleistenbasierte Autorensysteme: der Mozilla Popcorn Maker [Mozilla-Foundation 2012] (links) und das Konzept von *pipaciti* [Seidel 2008] (rechts)

Als Abstraktion des Zeitleistenparadigmas gibt es zwei bemerkenswerte Ausprägungen, die für sich als eigenständige Paradigmen erscheinen, es jedoch nicht sind. Das erste ist als *direct manipulation* betitelt und das zweite beruht auf Vannever Bushs Vision von *Memex*. In Anlehnung an Ben Shneidermans Konzept der „direct manipulation“ [Norman und Draper 1986; Shneiderman 1998] bilden so genannte *direct authoring interfaces* eine Ausprägung des Zeitleistenparadigmas, welches ohne metaphorische Repräsentationen der Medienobjekte auskommt [Seidel 2008]. Auf Grundlage eines audiovisuellen Basismediums lassen sich Medienobjekte direkt im Player annotieren, d.h. zeitlich und räumlich positionieren. Die Wiedergabe des Ausgangsmediums pausiert während der Bearbeitung der Objekte. Alle Änderungen werden unmittelbar in der Gestalt des Hypervideos sichtbar und sind beim Abspielen verfügbar. Der Autor eines Hypervideos kann damit sein eigenes Handeln mit der korrespondierenden Bildschirmanzeige kontrollieren und nach dem Schema „Test-Operate-Test-Exit“ (TOTE) [Khazaeli 2005, S. 202] verfahren. Eine beispielhafte Anwendung dieser Art stellt *Hyperfilm* [Srl 2007] dar.

*Memex* [Bush 1945] ist der Name einer, wenn auch nur fiktiven Maschine zur Suche und Repräsentation von Wissen. Sie dient dazu, auf Mikrofilm festgehaltene Bücher, Zeitungsartikel, Bilder, Schriftverkehr und Datensätze paarweise zu verknüpfen. Die jeweils indizierten Medien werden auf zwei Bildschirmen dargestellt. Parallel zum Lesen können Notizen und Kommentare, also Annotationen, mit Hilfe der Stifteingabe am Bildschirm ergänzt werden. Mittels eines Hebels erzeugt man Querbezüge („trails“) zwischen den aktuell dargestellten Seiten, die man dann mit einem weiteren Hebel verfolgen kann. Bush hatte damit im Jahre 1945 neben den Annotationen, auch das Prinzip der Hyperlinks entworfen und gleichzeitig eine aus zwei Zeitleisten bestehende Form der Autorenumgebung beschrieben, die eine adäquate Definition der Paarbeziehungen zwischen Quelle und Ziel eines Links erlaubt.

<sup>11</sup> Der Begriff Script bezieht sich hier auf Skriptsprachen im Bereich der Programmierung, nicht jedoch auf Lernskripte oder CSCL-Skripts.



Zusammenfassend kann man also sagen, dass Autorenumgebungen auf Basis von Zeitleisten bzw. dem *direct authoring* und Memex-Prinzip dem *bottom up* Ansatz nachkommen, während die übrigen Paradigmen eher ein *top down* Vorgehen implizieren. Das Paradigma der Zeitleiste, welches in der Film- und Videoproduktion seinen Ursprung hat, ist in den Autorenumgebungen am weitesten verbreitet. Dennoch finden sich Mischformen.

Neben den bislang betrachteten Einzelanwendungen gibt es im Kontext des E-Learnings ein weites Feld von Lernumgebungen, insbesondere LMS und Redaktionssysteme (CMS/LCMS). Systeme dieser Art integrieren Funktionen zur Erstellung, Durchführung und Organisation von Lernangeboten bzw. Kursen. Im folgenden Teilabschnitt werden daher Schnittstellen dieser Systeme beleuchtet, durch die Videoplayer und entsprechende Autorenumgebungen darin einbezogen werden können.

### 3.4.2 Integration in Lernumgebungen und Zusammenspiel mit Diensten im WWW

Die zu Beginn dieses Kapitels in Tab. 3.1 (S. 61) betrachteten klassischen LMS unterstützen kaum mehr als die Wiedergabe von Videos. Die für Hypervideos typische Interaktivität auf Mikro- oder Makroebene wird von diesen Systemen standardmäßig nicht unterstützt. Dennoch ist es nicht ausgeschlossen, Hypervideos in LMS zu integrieren. Im Folgenden werden technische Lösungsansätze für die Verbreitung und Integration von hypervideobasierten Lernressourcen kurz dargestellt und durch Beispiele belegt. Die Lösungsansätze betreffen Schnittstellen für Lerninhalte und Teilanwendungen sowie Austauschformate und semantische Auszeichnungen zum Wiederauffinden von Lernressourcen.

**Schnittstellen zur Integration und Distribution von Lerninhalten** Die Verbreitung von Lerninhalten im WWW kann im einfachsten Fall durch die Weitergabe von Links geschehen. Links lassen sich mit Hilfe von Nachrichtendiensten (z.B. E-Mail, twitter) sowie Sozialen Netzwerken versenden oder in Verzeichnissen (z.B. in *social bookmarking* Diensten) auflisten. Darüber hinaus können Medien und Anwendungen auch außerhalb ihrer Quelle verfügbar gemacht werden. Extern vorgehaltene Videos lassen sich etwa in einem Player wiedergeben, der Bestandteil der Lernumgebung oder eines *Social Networks* ist. Videoplayer können jedoch auch in ihrer Gesamtheit per *HTML IFrame* in eine Seite der Lernumgebung eingebettet werden (z.B. mit der *YouTube IFrame API*). Diese als trivial zu klassifizierende Form der Integration gestattet jedoch maximal die Rezeption von Hypervideo-Inhalten. Anwender haben keine Möglichkeit, Inhalte zu manipulieren oder miteinander zu kommunizieren.

**Schnittstellen zur Integration externer Teilanwendungen** Unter den Bezeichnungen Plugin, Extension oder Modul verfügen viele webbasierte LMS und CMS über Möglichkeiten, ihren Funktionsumfang durch Teilanwendungen zu erweitern. LMS/CMS bieten hierfür Programmierschnittstellen (API), mit deren Hilfe die Teilanwendungen Daten abfragen, modifizieren und speichern sowie auf bestimmte Routinen der Hauptanwendung zugreifen können. Die Teilanwendungen sind genau auf das System zugeschnitten, d.h. der Programmcode ist in der selben Programmiersprache geschrieben wie die Hauptanwendung. Sie sind deshalb mit einem relativ hohem Entwicklungsaufwand verbunden und nicht auf andere Systeme übertragbar.

Serviceorientierte Architekturen (SOA) bieten hierzu eine Alternative, indem ein externer Server die zu integrierenden Routinen ausführt und in Form eines Dienstes anbietet. Der Dienst wird z.B. in einem LMS/CMS registriert, um Daten über eine wohldefinierte Schnittstelle (API) abzurufen und zu senden. Dienste sind in sich geschlossen und plattformunabhängig, so dass ihre Integration in verschiedenen Hauptanwendungen mit geringem Entwicklungsaufwand möglich ist. Lucke und Tavangarian [2009] beschreiben wie sich Vorlesungsaufzeichnungen mittels SOA-Schnittstellen in verschiedene Lernumgebungen integrieren lassen.

Werden Daten, Darstellungen oder Programmroutinen mehrerer Quellen per Daten API in einer Darstellung zu einem neuen Dienst kombiniert, spricht man von einem *mashup*. Die Erstellung von *mashups* erfordert nicht unbedingt Programmierkenntnisse, wenn es wie beim *Mozillas Popcorn Maker* beispielsweise darum geht, Medien entlang einer Zeitleiste miteinander in Beziehung zu bringen [Mozilla-Foundation 2012].

**Interoperabilität von Lerninhalten** In den bisherigen Ausführungen wurde das Thema Interoperabilität in Bezug auf die Browser-Unterstützung von Video-Codec und Dokumentenformate betrachtet, die Beschreibungen des räumlichen und zeitlichen Layouts enthalten. Nun wird Interoperabilität in Bezug auf Lernressourcen betrachtet, um etwa Kurse in einem Format abzubilden, dass eine Zugänglichkeit, Austauschbarkeit und Wiederverwendung innerhalb unterschiedlicher webbasierter Lernumgebungen ermöglicht. Kurse bestehen im Allgemeinen aus einzelnen oder zusammengefassten Lernressourcen (z.B. Dateien oder URLs), Navigationsstrukturen inklusive der interaktiver Elemente sowie den Beiträgen der Lernenden.

Als Defacto-Standard für die Zusammenstellung webbasierter E-Learning-Inhalte hat sich das *Sharable Content Object Reference Modell* (SCORM) etabliert. SCORM ist Bestandteil vieler LMS sowie Software- und Content-Repositorien. Auch einfache Video-Inhalte können via SCORM in LMS repräsentiert werden. Ferner unterstützen einige LMS auch Inhalte im AICC-Format<sup>12</sup>, in dem Schnittstellen und Regeln für CBTs spezifiziert sind. Mit SCORM und AICC ist es auch möglich, Videos in HTML-Seiten einzubetten und zwischen den Seiten zu navigieren. Die Verfügbarkeit einzelner Seiten kann zudem an verschiedene Bedingungen geknüpft sein (z.B. ein Verfallsdatum oder eine maximale Anzahl an Seitenaufrufen). Interaktive Videos im Sinne einer Mikrointeraktivität lassen sich per JavaScript innerhalb der HTML-Dokumente realisieren.

Zusammenfassend ist zu bemerken, dass die genannten Standards nur Angaben zur Struktur von Dateien bzw. Dateiinhalten erfassen, ohne dabei Beiträge und Kommunikationsprozesse der Lernenden zu berücksichtigen. Von den Standards profitieren vorrangig Lehrende, die beispielsweise eine Kursstruktur in einem kommenden Semester wiederverwenden möchten, wohingegen Lernende ihre darin eingebrachten Lernleistungen (z.B. Diskussionsbeiträge, Testergebnisse oder Wiki-Artikel) nicht sichern oder transferieren können. Neuere Initiativen wie *Tin Can API* adressieren dieses Defizit und helfen Studierenden ihre auf viele Systeme (und Organisationen) verteilten Lernaktivitäten und Ergebnisse zu bündeln, d.h. für sich persönlich zu sichern, zu kontrollieren und in andere LMS zu transferieren.

---

<sup>12</sup>Dieser Standard wurde anfänglich für die Luftfahrtindustrie entwickelt. Das *Aviation Industry Computer-Based Training Committee* zeigt sich für die Weiterentwicklung verantwortlich.

Anders als im E-Assessment, wo seit langem Standards für Überprüfungsfragen, Tests und Bewertungsschemata bestehen (z.B. *QTI*, *GIFT*), gibt es für die Produktion und den Einsatz von Lernvideos bislang noch keine Empfehlungen, um einen systemübergreifenden Einsatz zu ermöglichen. Vielversprechend ist jedoch die vom *IMS Global Learning Consortium* ins Leben gerufene Initiative zur Entwicklung eines *OpenVideo Standards*<sup>13</sup>. Der Standard soll Workflows für die Produktion, Datenhaltung und Distribution von Videos enthalten und in verschiedene LMS bzw. videoverarbeitende Systeme integriert werden.

**Semantische Auszeichnung für Suchmaschinen** Das Anliegen, videobasierte Lernressourcen mit semantischen Annotationen zu versehen, ist vor allem dadurch begründet, die Informationen im WWW für Dritte wieder auffindbar zu machen. Suchmaschinenbetreiber empfehlen Web-Inhalte in einer bestimmten Form aufzubereiten. Diese Art der Aufbereitung wird allgemein als *Search Engine Optimization* (SEO) bezeichnet, da die Qualität der Auszeichnungen mit über den Rang in der Ergebnisliste einer Suchanfragen entscheiden kann. Dies ist auch für Lehrende und Lernende nicht unerheblich, wenn sie ihre elektronischen (Video-)Lernmaterialien für eine nicht näher bestimmte Gruppe von Lernenden zugänglich und auffindbar machen wollen. Google, Yahoo und Microsoft einigten sich darauf für verschiedene Arten von Web-Inhalten, darunter auch Videos, Metadaten-Schemata<sup>14</sup> zu definieren<sup>15</sup> [Schonfeld und Shivakumar 2009]. Diese so genannten Sitemap-Protokolle basieren auf XML und enthalten beispielsweise Informationen über die Art und Struktur des Inhalts sowie die Urheber, mögliche Verwertung und Update-Zyklen.

Insbesondere für die Verbreitung und Integration von frei verfügbaren hypervideobasierten Lernressourcen im Sinne von *Open Educational Resources* sind basale *Sitemaps* für Suchmaschinen eine notwendige Voraussetzung.

## Zusammenfassung

Im zurückliegenden Kapitel wurden Videos sowie Hypervideos aus einer technischen Perspektive betrachtet und hinsichtlich spezifischer Herausforderungen erläutert. Als eine Teilmenge von Hypermedia werden Hypervideos in Dokumenten spezifiziert, die räumliche und zeitliche Ordnungsbeziehungen der Inhalte manifestieren. Im Kontext des E-Learnings unterscheiden sich Videolernumgebungen gegenüber gewöhnlichen LMS. Diese Lernumgebungen können LMS jedoch um passende Werkzeuge zur Unterstützung von Lernaktivitäten und zur Bearbeitung videografischer Lernmedien erweitern.

In zweiten Abschnitt wurde das Konzept des *Human Video Interface* begrifflich eingeführt und hinsichtlich seiner möglichen Gestaltungsformen innerhalb des Gestaltungsraums abgegrenzt. Der Gestaltungsraum beinhaltet zeitabhängige bzw. zeitbezogene Elemente innerhalb und außerhalb des Videos sowie zeitgenaue Hyperlinks als Navigationselemente im *hypermedia space* und Zeitleisten als Projektionsfläche des zeitlichen Layouts. Diese Charakteristika

<sup>13</sup>Siehe <http://www.imsglobal.org/activity/openvideo> (abgerufen am 21.01.2016).

<sup>14</sup>Im Gegensatz dazu beziehen sich andere Metadatenstandards für audiovisuelle Medien, wie z.B. *Dublin Core*, *MPEG 7*, *MPEG 7 Detailed Audiovisual Profile* oder *RSS-TV* [Kürsten 2009], lediglich auf Dateien, anstatt auf deren Beziehungen zueinander.

<sup>15</sup>Siehe: <http://www.sitemaps.org/schemas/sitemap/0.9> (abgerufen am 15.02.2016).

stellen potentielle Interaction Design Patterns dar, die sich anhand von Beispielen weiter ausdifferenzieren lassen.

Aus den Besonderheiten der zeitlichen Kontinuität von Hypervideos und der ihnen innewohnenden Netzstruktur ergeben sich gesonderte Herausforderung an die Usability (siehe Abschnitt 3.2.2, S. 74). Diese Herausforderungen konnten in Guidelines überführt werden (siehe Anhang S. 261), die sich als Checkliste im Rahmen der Konzeption, Entwicklung und Evaluation entsprechender Systeme einsetzen lassen. Nichtsdestotrotz stellen heuristische Hilfsmittel dieser Art nur einen ersten Schritt auf dem Weg zur Verbesserung der Usability von Hypervideo-Systemen dar.

Aus den Ausführungen im dritten Abschnitt des Kapitels wird deutlich, dass die technische Umsetzung von Hypervideos keine triviale Angelegenheit darstellt. Die Ursache dafür ist die ungenügende Integration zeitbasierter Medien in den aktuellen HTML5-Standards des W3C. In Folge findet man viele der grundlegenden Mechanismen nicht in den Implementierungen der aktuellen Web-Browser. Um leistungsfähige Hypervideo-Systeme für das WWW entwickeln zu können, wurden sieben Anforderungen ermittelt, die im Fortgang dieser Arbeit für die Entwicklung eines Frameworks für interaktive Videos (siehe VI-TWO im Kapitel 6, S. 215) relevant sind.

Im vierten Abschnitt wurden Vorgehensweisen bei der Produktion von Hypervideos erläutert und in Beziehung zu Paradigmen von Autorensoftware gebracht. Eine Gegenüberstellung von Autorenumgebungen ergab, dass die meisten Anwendungen auf dem Zeitleisten-Paradigma beruhen. Die Entwicklung von nicht-linearen Informationssystemen sowie die kollaborative Bearbeitung der Inhalte unterstützen nur wenige Systeme. Im Hinblick auf das Lernen mit Videos wurden im zweiten Teilabschnitt Möglichkeiten eruiert, um Hypervideos in bestehende LMS oder CMS zu integrieren oder in Form von *web services* nutzbar zu machen. Da Content-Formate wie SCORM lediglich in LMS eingesetzt werden können und Plugins für jedes System neu entwickelt werden müssen, stellen SOA-Schnittstellen sowie eingebettete Inhalte eine Alternative dar.

**Kernpunkte**

- Stellen Videos in einer Lernumgebung die primäre Lernressource dar und werden Lernende bei ihren Lernaktivitäten (einschließlich der Bearbeitung) durch Werkzeuge für diese kontinuierlichen Medien unterstützt, spricht man von einer Videolernumgebung.
- Hypervideos zeichnen sich durch Steuerungs- und Auswahlelemente, eine nicht-lineare Struktur, alternative Wiedergabepfade, variable Szenenabfolge sowie die Möglichkeit, Zusatzinformationen zu ergänzen und zu modifizieren, aus.
- Videolernumgebungen stellen eine spezielle Form von Hypervideo-Anwendungen dar.
- Für die meisten Formen von Hypervideos genügt eine annotationsbasierte Spezifikation des Hypervideo-Dokuments.
- Unter *Human Video Interface* versteht man den Teil eines Videosystems, mit dem Anwender mit dem Video interagieren und kommunizieren können.
- Autorenumgebungen für Videos und Hypervideos basieren zumeist auf mehrspurigen Zeitleisten, wenngleich kollaborative Nutzungsformen sowie die Einbeziehung nicht-linearer Navigationsmöglichkeiten eine Ausnahme darstellen.



## Kapitel 4

# Interaction Design Patterns für Videolernumgebungen

Ausgehend von den Betrachtungen zum Lernen mit Videos in Kapitel 2 sowie den in Kapitel 3 erläuterten technischen Rahmenbedingungen von Videolernumgebungen und Hypervideos wurde der zu gestaltende Gegenstand einer Videolernumgebung definiert und hinreichend eingegrenzt. Im Zuge dieser Betrachtungen traten viele funktionale Merkmale hervor, die bei der Konzeption und Entwicklung eine Rolle spielen können. Ausgangspunkt für die Gestaltung und Entwicklung von Videolernumgebungen ist die Schnittstelle zwischen Mensch und Video, die sich durch vielfältige Formen und Funktionen ihrer Elemente auszeichnet. Als Gestaltungslösungen stellen diese Elemente keine singulären Erscheinungen dar, sondern wiederkehrende Lösungen mit verschiedenen Ausprägungen und einem gemeinsamen Kern. Beispielsweise ermöglichen eine Reihe von Videoplayern die Anpassung der Wiedergabegeschwindigkeit nach oben und unten. Während diese Funktion in allen Playern gleich ist und somit den Kern darstellt, finden sich ganz unterschiedliche Ausprägungen der Steuerungselemente (z.B. Drop Down Menü, Schieberegler, +/- Schaltflächen) sowie unterschiedliche Wertebereiche und Abstufungen der Geschwindigkeit. Noch größer ist die Varianz der Lösungen bezüglich des Designs (Form, Größe, Farbe, Icon, Schriftart).

Die Frage nach der Gestaltung von Videolernumgebungen wird dabei nicht als Ganzes für eine oder mehrere Lernumgebungen behandelt, sondern für die Betrachtung einzelner Elemente der Lernumgebungen aufgeschlüsselt. Mit Hilfe der Mustertheorie von Alexander [1979] wird die Gesamtlösung einer Videolernumgebung als eine Menge von bewährten Teillösungen für wiederkehrende (Teil-)Probleme im jeweiligen Anwendungskontext betrachtet. Diese Lösungen für ein Problem im Kontext bezeichnet man als Entwurfsmuster (engl. Design Pattern), oder kurz als Muster oder Pattern. Die Vorgehensweise entspricht dabei dem *Divide-and-Conquer*-Ansatz, wobei die resultierenden Beschreibungen der Patterns nicht isoliert von einander existieren, sondern als eine strukturierte und miteinander verknüpfte Sammlung von Patterns entwickelt werden. Eine solche Sammlung von Mustern bezeichnet man als Muster-sprache (engl. Pattern Language). Die Bezeichnung als Sprache rührt einerseits daher, dass die Entfaltung von Mustern als strukturelle Formensprache verstanden werden kann und andererseits Muster als ein Vokabular in der Kommunikation von Laien und Experten dienen.

Ziel dieses Kapitels ist es, bewährte Gestaltungslösungen des *User Interface* von Videolernumgebungen aus einer Vielzahl von Einzelfällen zu identifizieren und ihre Notwendigkeit durch Angabe der dafür ursächlichen Probleme im jeweiligen technischen oder didaktischen Kontext zu begründen. Die Zielsetzung behandelt somit die Forschungsfrage FF 1 nach bewährten Lösungen für häufig auftretende Probleme (Design Patterns) hinsichtlich der Gestaltung des *User Interfaces* von Videolernumgebungen zu suchen.

Der ersten Teil des Kapitels beinhaltet eine Einführung in grundlegenden Begrifflichkeiten der Mustertheorie. Neben der Erläuterung von basalen Bestandteilen und Eigenschaften von Patterns und Pattern Languages erfolgt eine Einordnung in und Abgrenzung zu verwandten Methoden im Bereich der Mensch Computer Interaktion (MCI).

Im zweiten Teil des Kapitels wird eine systematische Vorgehensweise vorgestellt, um wiederkehrende Gestaltungslösungen im *User Interface Design* von Videolernumgebungen zu identifizieren und daraus Interaction Design Patterns sowie eine Pattern Language zu entwickeln. Kernbestandteil dieser Methode ist eine empirische Inhaltsanalyse von 121 Videolernumgebung hinsichtlich ihrer funktionalen Merkmale im *User Interface* sowie eine Analyse verwandter Pattern Languages zur Identifikation struktureller und inhaltlicher Ähnlichkeiten.

Als Ergebnis dieser Methode steht eine Pattern Language mit 45 Patterns, die im letzten Teil des Kapitels vorgestellt und hinsichtlich ihrer Repräsentationsformen und Einsatzmöglichkeiten erläutert wird.

## Einführung in Design Patterns

In diesem Abschnitt wird der methodische Ansatz zur Beschreibung wiederkehrender Lösung für häufig auftretende Probleme in Gestalt von Patterns eingeführt. Der erste Teilabschnitt ist den Bestandteilen der in sich strukturierten Pattern-Beschreibungen gewidmet. Insbesondere gilt es dabei, den Typus der in Patterns zum Ausdruck kommenden Gestaltungsprobleme zu durchdringen und eine Vorstellung von der Beschaffenheit der Lösungen zu bekommen. Im zweiten Teilabschnitt werden Patterns im Verbund von Pattern Languages betrachtet. Das Konzept der Pattern Language besticht durch seine Ordnung in Form von Kategorisierungen und Querverweisen, welche die fragmentarischen Einzellösungen in einem Zusammenhang rücken, um damit komplexeren Gestaltungsaufgaben in der Praxis zu begegnen. Inwieweit dazu auch andere Gestaltungsrichtlinien geeignet sind, wird im dritten Teilabschnitt diskutiert.

Zunächst aber geht es um die Frage nach den praktischen Nutzen und den Potentialen von Patterns. Allein im Bereich der Informatik sind derzeit über 100 einschlägige Bücher über Design Patterns auf dem Markt [Leitner 2015]. In vielen Teildisziplinen, insbesondere der Software Entwicklung und dem Interaction Design, bilden sie einen festen Bestandteil des Curriculums von Informatikstudiengängen. Der Ansatz findet aber auch in der Praxis von Unternehmen Resonanz. Patterns haben dabei mehrere Funktionen. Sie sind ein Mittel der Wissensvermittlung, Kommunikation und des Wissensmanagements. In der folgenden Auflistung werden diese und weitere Potentiale im Einzelnen erläutert.



- **Lernen:** Unterstützung für Personen, die über wenig Erfahrung in einem Anwendungsgebiet oder mit einer bestimmten Plattform verfügen. Insbesondere Software Entwickler, aber auch Designer sind mit dem Konzept von Design Patterns vertraut [Kunert 2009, S. 66f].
- **Kommunikationsmittel:** Patterns sind unkompliziert geschrieben und gut lesbar für Designer, Entwickler und andere Beteiligte und dienen deshalb als ein Kommunikationsmittel [Seffah und Taleb 2012]. Im Sprachgebrauch von Experten eines Fachgebiets werden die Namen von Patterns als Synonym der darin beschriebenen Lösung genutzt. Dies erleichtert die Kommunikation der beteiligten Personen, denen die mit einem Pattern-Namen verbundenen Lösungen vertraut sind [Kunert 2009, S. 66f].
- **Transdisziplinäre Wissensvermittlung:** Sie erlauben es, Wissen und Erfahrungen aus verschiedenen (Teil-)Disziplinen in einer einheitlichen Form zu beschreiben. Zwischen Aspekten einzelner Bereiche werden insbesondere Bezüge hergestellt, die für die Umsetzung relevant sind. Außerdem repräsentieren sie Erkenntnisse aus Gestaltungsprozessen aus verschiedenen Perspektiven [Seffah und Taleb 2012].
- **Wissensmanagement:** Patterns stellen ein relativ intuitives Hilfsmittel dar, um Erkenntnisse aus Gestaltungsprozessen und „good practice“ zu dokumentieren [Seffah und Taleb 2012]. Als Entscheidungshilfe zur Auswahl erprobter Lösungen helfen sie Anwendern, Lösungen für ihre spezifischen Anforderungen im jeweiligen Anwendungskontext zu finden [Kunert 2009, S. 66f]. Unterschiedliche Produkte und Darstellungen (z.B. Bücher, Artikel, Kartendecks, Videos, Aufkleber, Online-Kataloge und Datenbanken) unterstützen die Nutzung der Patterns.
- **Empirische Belege:** Patterns werden i.d.R. aus erfolgreichen Experimenten und bekannten Beispiellösungen abstrahiert. Sie repräsentieren somit weder experimentelle Prototypen, noch ungeprüfte Theorien, sondern die Essenz funktionierender Lösungen.
- **Offenheit:** Patterns zeigen Wege auf, anstatt sie vorzugeben. Chancen und Grenzen einzelner Lösungen werden als ein Kompromiss in Anbetracht ihrer Alternativen und Wechselwirkungen erörtert. Sowohl die Textform der Patterns, als auch die Querbeziehungen in einer Pattern Language sind weder als abgeschlossene, noch als endgültige Wissensrepräsentationen anzusehen. In größeren Unternehmen (z.B. Crumlish und Malone [2009]; Infragistics [2013]; Rising [1999]; Yahoo! [2012]) ist es nicht unüblich, Patterns für individuelle Nutzungsszenarien anzupassen oder mehrere Pattern Languages miteinander zu kombinieren. In Anbetracht neuer Entwicklungen, Erkenntnisse und Erfahrungen bedürfen Patterns jedoch einer regelmäßigen Aktualisierung und Überarbeitung.

Die obigen Potentiale lassen bereits erkennen, dass verschiedene Zielgruppen unter Umständen einen ganz unterschiedlichen Nutzen aus der Verwendung von Design Pattern ziehen können. Die in dieser Arbeit erstellten Design Patterns richten sich an vier Zielgruppen, die jeweils in die Entwicklung von Videolernumgebungen involviert sind. Zunächst dienen die Patterns dazu, *User Interface Designer* für zeitbasierte interaktive Medien zu sensibilisieren und sie auf Besonderheiten zeitabhängiger Ereignisse und zeitbezogener Navigationselemente aufmerksam zu machen. Die zweite und dritte Zielgruppe stellen Software Entwickler sowie Anbieter von CMS/LMS dar. Diese beiden Gruppen sollen dazu angeregt werden, Web Videos nicht länger als in sich geschlossene und kaum zugängliche Medien zu verstehen. Die Patterns ver-

deutlichen, wie bestehende Player und CMS/LMS um zeitabhängige Inhalte sowie zeitbezogene Interaktionen erweitert werden können. Im Hinblick auf ein möglichst zeitloses Interaktion Design enthalten die Patterns allerdings weder Code-Beispiele, noch konkrete plattform-, browser- oder frameworkbezogene Implementierungshinweise. Des Weiteren können Lehrende – als eine vierte Zielgruppe – die Patterns nutzen, um bestehende Videolernumgebungen oder bestimmte Gestaltungsmerkmale derselben zu bewerten und miteinander zu vergleichen. Patterns zeigen zudem einen Weg auf, Videolernumgebungen zu entwerfen und hinsichtlich individueller Anforderungen zu konzipieren. Insbesondere die in den Patterns zum Ausdruck kommenden anschaulichen Beispiele und Querverweise zwischen den Lösungen repräsentieren den Gestaltungsspielraum hinsichtlich der in der Praxis erprobten Lösungen.

Wie relevant HCI Patterns für die genannten Zielgruppen sind, wurde noch nicht umfassend untersucht. Die praktische Relevanz von HCI Patterns für den Arbeitsalltag von Forschern und Software Entwicklern haben Kruschitz und Hitz [2010] in einer Studie erhoben. 171 von 286 befragten Personen (knapp 60 %) nutzen HCI Patterns. Über 70 % der Pattern-Nutzer gaben an, dass Patterns dazu beitragen, den Gestaltungsprozess zu verbessern. Als Begründung führten sie verschiedene Einsatzszenarien an, z.B. die Verwendung von Patterns als Checkliste, die Ermittlung von umfassenderen Kontextinformationen (im Vergleich zu anderen Guidelines), Vermeidung von verbreiteten Fehlern sowie die Vereinheitlichung von Prozessen innerhalb großer Projekte mit vielen Beteiligten.

Neben den angeführten Nutzungspotentialen gibt es auch eine Reihe von Kritikpunkten, die mit dem methodischen Ansatz und dem praktischen Einsatz zusammenhängen. Nicht selten tendieren Novizen dazu, Patterns als Blaupause zu verwenden. Durch die vorgegebene Lösung sind Anwender geneigt auf die Suche nach alternativen Lösungen zu verzichten, obwohl diese unter Umständen besser sein können. Auf diese Weise manifestieren sich Patterns durch ihre fortwährende Anwendung. Je mehr sich eine Lösung in der Praxis verbreitet, desto relevanter erscheint das Pattern. Auch durch den Einsatz von Patterns als Lernmittel für Novizen verstärkt sich ihre scheinbare Relevanz. Andersherum kann man aber auch argumentieren, dass eine häufige Anwendung von Patterns durch unabhängige Experten nur die Gültigkeit der darin verbrieften Lösung unterstreicht und bestätigt.

Ein weiterer methodischer Kritikpunkt bezieht sich auf den limitierten Gültigkeitsbereich von Patterns. Deutlich wird dies anhand der „Gang of Four“ Patterns [Gamma et al. 1995], die gemeinhin als Design Patterns für die objektorientierte Programmierung angesehen werden, obwohl die Lösungen eigentlich nur für eine bestimmte objektorientierte Programmiersprache (C++) bestimmt sind. Für einige dieser Patterns besteht in anderen objektorientierten Sprachen (z.B. JAVA) überhaupt keine Notwendigkeit. Patterns adressieren und lösen teilweise Probleme, die nur in ihrem spezifischen Kontext existieren (z.B. das VISITOR-Pattern in JAVA). Im HCI-Bereich ist der Kontext oft durch die Plattform geprägt (z.B. Desktop-PC, Smartphone, Spielkonsole, TV-Gerät), da die Ein- und Ausgabegeräte die Gestaltungslösungen maßgeblich determinieren. Das Fehlen allgemeingültiger, d.h. plattformunabhängiger Patterns, führen einige Autoren darauf zurück, dass die Disziplinen Software Engineering und HCI im Vergleich zur Architektur und dem Bauwesen noch zu jung sei (siehe z.B. Gamma et al. [1995]).

In eine ähnliche Richtung stößt die Kritik, Pattern würden als „best practice“ oder „good practice“ einem Dogma gleichgesetzt. Kohls [2014], Kohls und Panke [2010] u.a. plädieren deshalb dafür, Patterns als Hypothesen aufzufassen, für die ein empirischer Beleg erst noch zu erbringen ist.

Aus Anwendersicht wird gelegentlich der Umfang an Text bemängelt, der die Nutzung und die Suche nach passenden Lösungen erschwert und viel Zeit erfordert. Wenngleich diese Kritik für die oftmals mehrere hundert Seiten umfassenden Pattern-Bücher nachvollziehbar sein mag, so scheint das Problem in einigen Online-Pattern-Repositories dank Volltextsuche, Entscheidungshilfen und Übersichtsdarstellungen bereits gelöst. Anwender von HCI-Patterns überfliegen lediglich die Problemstellungen, scannen die Lösung und schauen auf die Bilder. Teilweise werden nur die Abbildungen rezipiert.

Angesichts dieser Kritik, werden im weiteren Verlauf noch einen Blick auf ähnliche bzw. alternative Ansätze zur Vermittlung von Gestaltungsprinzipien (siehe Teilabschnitt 4.1.3, S. 104). Zuvor bedarf es jedoch einer genaueren Einführung in die Bestandteile und Eigenheiten von Patterns und Pattern Languages.

#### 4.1.1 Design Patterns

Nachdem im letzten Abschnitt einige Beweggründe zur Nutzung von Patterns erläutert wurden, geht es im Folgenden um das Wesen, die Bestandteile und Eigenschaften von Design Patterns. Zunächst wird der Begriff der Design Patterns wie folgt definiert:

**Definition 4.** *Design Patterns beschreiben verbreitete Lösungen für wiederkehrende Gestaltungsprobleme in einem Kontext.*

Zu dieser Definition sei angemerkt, dass im Rahmen dieser Arbeit die Bezeichnungen Muster, Entwurfsmuster, Pattern und Design Pattern als synonyme Begriffe verwendet werden.

Patterns stellen eine strukturierte, textliche und grafische Beschreibung dar, die mindestens das Problem und die Lösung sowie den Kontext von beiden enthält. Als Musterinstanz bezeichnet man ein zur Anwendung gelangtes Muster. Eine Instanz besitzt eine individuelle Gestaltungslösung, wobei der Kern der Lösung dem Muster entspricht.

Grundlegend haben Muster den Charakter deskriptiver Modelle, indem sie bekannte Problemstellungen und Wechselwirkungen beschreiben bzw. von ihnen abstrahieren [Alexander 2002, S. 344], [Martin et al. 2001, S. 50]. Mit der Modellentwicklung geht das Erkenntnisinteresse einher, komplexe Sachverhalte zu vereinfachen und in Form einer Musterbeschreibung nachvollziehbar zu veranschaulichen. Hinsichtlich ihrer Zweckbestimmung handelt es sich jedoch um präskriptive Modelle [Alexander 2002, S. 344] [Martin et al. 2001, S. 42], da sie Lösungen für Probleme anbieten. Alexander führt die präskriptive Komponente auf die Entdeckung zurück, mustergültige Systeme künstlich erschaffen zu können [Alexander 2002, S. 344]. Der sich daraus ableitende normative Charakter bezieht sich auf ein übergeordnetes, mehrere Muster betreffendes Ziel.

Ein Muster setzt sich aus einer Ordnung wohldefinierter Elemente zusammen. Diese Struktur ist invariant für alle Muster innerhalb einer Sammlung oder Pattern Language. Unterschiede bestehen zwischen unterschiedlichen Pattern Languages hinsichtlich der Bezeichnung, Reihenfolge und dem Inhalt der Elemente. Eine vergleichende Darstellung dieser unterschiedlichen Ausprägungen findet sich bei Köhne [2005, S. 129ff]. Die ursprüngliche Form des alexandrischen Patterns besteht aus den Elementen: Name, Reifegrad, Kontext, Problem, Wechselwirkung/Forces, Lösung, Konsequenzen, Beispiele Alexander [1979]; Alexander et al. [1977]. Diese sollen im Folgenden näher erläutert werden.

**Name** Der Name eines Musters gilt als Synonym für die darin beschriebene Lösung, weshalb eine kurze und prägnante Bezeichnung gewählt werden sollte.

**Reifegrad** Alexander et al. [1977] drückt durch die Angabe von ein bis drei Sternchen aus, wie überzeugt sie von dem Pattern sind. Einige Autoren von Entwurfsmustern folgen diesem Ansatz (z.B. Köppe [2012b]). Da es sich jedoch um eine subjektive und somit auch fehleranfällige Einschätzung handelt, fehlt dieses Element in vielen Pattern Languages. Beispielsweise verzichten [Crumlish und Malone 2009; Kunert 2009; Tidwell 2011; van Welie und Veer 2003] im Bereich der Mensch Computer Interaktion und [Avgeriou et al. 2003a; Bauer und Baumgartner 2012; Finlay et al. 2010; Köhne 2005] in ihren E-Learning-Patterns auf die Quantifizierung, wie sicher sie sich eines Patterns sind. Nur wenige Autoren koppeln den Wert an eine feste Messgröße wie z.B. die Anzahl an erfolgreichen Anwendungsbeispielen [Bauer und Baumgartner 2010].

**Kontext** Der Kontext eines Problems ist ein Sammelbegriff für die Beziehungen zu anderen Gegenständen bzw. Eigenschaften. Der Kontext schränkt den Anwendungsbereich eines Musters ein [Kohls und Panke 2010]. Je weiter der Kontext gefasst ist, desto größer ist das vermeintliche Anwendungsfeld und desto umfangreicher sollte sich die Menge an Belegen aus den möglichen Anwendungsfeldern darstellen. Ist der Kontext im Gegensatz dazu zu eng definiert, schränkt dies die Anwendungsmöglichkeiten ein. Wichtig anzumerken ist, dass der Kontext auch nach der Anwendung der Lösung unverändert bestehen bleibt.

Ändert sich hingegen der Kontext einer Anwendung, so kann eine Lösung nicht mehr als passend gelten. Die Lösung ist damit keine Lösung mehr, so lange sie nicht an den veränderten Kontext angepasst wird. Grundlegend ist damit auch klar, dass ein Kontext nicht zweimal existieren kann und folglich Lösungen immer etwas angepasst werden müssen. Die Beschreibung des Kontextes als Teil des Musters muss damit hinreichend abstrakt erfolgen. Im Gegenzug ist nicht auszuschließen, dass Lösungen analog auf einen anderen Kontext anwendbar sind. [Kohls und Uttecht 2009]

**Wechselwirkungen / Forces** Forces, zu deutsch Wechselwirkung, bilden den Kern einer Mustersprache. Durch sie kommen Anforderungen und Beschränkungen der beschriebenen Lösung innerhalb des Kontextes zum Ausdruck. Somit trägt die Beschreibung von Wechselwirkungen zur Begründung der Lösung bei [Schümmer 2010; Wellhausen und Fiesser 2012]. Während Muster Lösungen bieten, machen Forces auf Probleme aufmerksam. Jede Komponente der Lösung ist deshalb mit einer oder mehreren Forces verknüpft. Einige Wechselwirkungen

gleichen sich durch die Anwendung der Lösung aus, während andere in Konflikt zueinander stehen. Konflikte müssen durch andere Muster aufgelöst werden.

Nicht einfach zu beantworten ist daher die Frage, ob alle im Kontext relevanten Wechselwirkungen in der Musterbeschreibung erfasst wurden. Buschmann und Henney [2008] geben dahingehend eine Hilfestellung, indem sie zwischen drei Ebenen unterscheiden:

1. Wechselwirkungen innerhalb der Mustersprache als zentrale Wechselwirkungen innerhalb des Geltungsbereichs der Mustersprache. Jeder dieser Wechselwirkungen begegnet man mit einem gesonderten Pattern.
2. Individuelle Wechselwirkungen innerhalb des Musters, die sich aus verschiedenen Umsetzungsvarianten einer Lösung ergeben.
3. Domänenspezifische Wechselwirkungen sowie solche, die auf externe Einflüsse (z.B. technische Limitationen) zurückzuführen sind.

Im Interaction Design gibt es wiederkehrende Forces, wie z.B.

- den räumlich begrenzten Interaktionsbereich,
- Grenzen der visuellen und akustischen Wahrnehmungsfähigkeit,
- die entstehende kognitive Last aufgrund einer hohen Informationsdichte,
- Limitationen der computervermittelten Kommunikation,
- Gestaltgesetze.

In Bezug auf audiovisuelle Medien ergeben sich zudem Forces aus der Flüchtigkeit zeitabhängiger Informationen sowie aus der kognitiven Last aufgrund einer gleichzeitigen Darstellung von visuellen und auditiven Informationen (vgl. Abschnitt 2.2, S. 36).

**Problem** Muster enthalten eine Lösung für ein Problem im Kontext. Nachdem dem Kontext stellt sich die Frage nach dem Wesen des Problems, für das eine Lösung angeboten wird. Abstrakt könnte man die Aufgabe, eine Videolernumgebung zu konzipieren und zu entwickeln, als eine Problemstellung ansehen. Offensichtlich handelt es sich dabei um einen schöpferischen Akt oder eine gestalterische Aufgabe. Aufgaben dieser Art haben in gewissen Disziplinen einen außerordentlich hohen Stellenwert. So stellt das Lösen von Gestaltungsproblemen für Architekten beim Entwurf von Gebäuden, für Software Entwickler bei der Entwicklung von Systemarchitekturen und für Lehrende bei der Konzeption und Entwicklung von Lehr-Lern-Szenarien eine typische Aufgabe dar [Alexander 2003]. Bevor Spezifika von Gestaltungsproblemen näher erläutert werden, ist es erforderlich auf die von Zymbardo und Gerrig [2009, S. 302-303] genannte Bestandteile eines Problems einzugehen:

1. Anfangszustand,
2. Zielzustand,
3. Menge an Operatoren, um von 1. nach 2. zu gelangen.

Zusammen bilden diese drei Elemente den Problemraum. Ein Problem entsteht, genau dann wenn bei der Ausführung einer Handlung eine Hürde auftritt. Dies kann im Anfangszustand durch unvollständige Informationen oder eine unbefriedigende Zustandslage bedingt sein. Auch der Zielzustand muss nicht zwingend durch gesuchte Informationen oder eine Vorstellung einer Lösung gekennzeichnet sein. Gleiches gilt für die Operatoren beim Zustandsübergang. Sobald

auch nur ein Element des Problemraums undefiniert ist, spricht man in der Gestaltpsychologie von einer „Lücke“ im Handlungsablauf [Wertheimer 1964, S. 58].

Simon [1973] bezeichnet diese Form von Problem als unstrukturierte („ill structured“), d.h. unvollständig definierte Probleme. Im Gegensatz dazu ist ein wohl strukturiertes Problem anhand eines eindeutigen Kriteriums überprüfbar. Um ein Kriterium zur Bewertung einer Lösung eines unstrukturierten Problems zu finden, müsste der Problemraum klar spezifiziert sein [Simon 1973]. Für die Konzeption einer Videolernumgebung hieße dies, alle relevanten Operatoren, d.h. alle Design-Artefakte, Interaktionselemente, Arbeitsprozesse sowie deren Organisation in Betracht zu ziehen. In Konsequenz ließe sich der Problemlösungsprozess automatisieren. Typische Gestaltungsprobleme dieser Art sind jedoch nicht strukturiert, sie lassen sich aber teilweise strukturieren [Simon 1973].

Ein Lösungsansatz zur Strukturierung des Problemraums besteht in der Zerlegung des Problems in mehrere Teilprobleme mit überschaubaren Problemräumen. Zymbardo und Gerrig [2009, S. 303] definieren die Zerlegbarkeit als eine wesentliche Eigenschaft eines Problems. In der alexandrinischen Mustersprache kommt dies in der immanenten Hierarchie und Relation von Mustern zum Ausdruck. Muster schließen mehrere andere Muster ein und sind selbst wieder ein Teil eines übergeordneten Musters. Das Gesamtproblem der Gestaltung einer Videolernumgebung lässt sich also durch eine Vielzahl an Mustern aufteilen, die jeweils Teilprobleme mit den korrespondierenden Teillösungen repräsentieren. Daraus ergeben sich unterschiedliche Detailgrade von Problemen. Van Welie und Veer verweisen in dem Zusammenhang auf eine Hierarchie von Größenordnungen eines Problems („scale hierarchy“ of problems“) [van Welie und Veer 2003]. Und auch Alexander et al. [1977] hatten eine Hierarchie von Problemen in der Architektur von Städten, Gebäuden und in der Konstruktion vorgesehen. Trotz der Zerlegung in Teilprobleme lassen sich die durch Muster beschriebenen Probleme nicht wohl strukturieren. Die Generizität der Muster bedingt Freiheitsgrade in der Umsetzung einer Lösung, die immer noch auf einem, wenn auch spezifischeren Gestaltungsproblem beruhen.

Eine Klasse von Problemen lässt sich auf ein Muster reduzieren und in das durch ein Muster beschriebene Problem überführen, sofern der Kern des Problems übereinstimmt. Muster bilden somit Problemklassen.

Kohls [2014, S. 38] unterscheidet in Bezug auf Patterns drei Arten oder Level von Problemen: Erstens das Kernproblem, welches aufgrund drohender negativer Effekte gelöst werden muss, zweitens das Finden einer Lösung für das Kernproblem und drittens das Problem die Lösung erfolgreich umzusetzen. Patterns beschreiben prinzipiell lösbare Probleme, was durch bekannte Lösungsinstanzen in der Musterbeschreibung zum Ausdruck kommt. Dies schließt jedoch nicht aus, dass es abgesehen von den im Pattern inbegriffenen Lösungsvarianten auch noch gänzlich alternative Lösungen gibt.

Das Finden eines guten Lösungswegs stellt nicht nur ein gesondertes Problem dar, sondern bildet selbst ein Pattern oder „process pattern“ [Kohls 2014, S. 40]. Auch hier kann es mehrere Lösungen geben, die sich beispielsweise durch unterschiedliche Methoden, Abläufe oder Werkzeuge voneinander unterscheiden. Vor dem Hintergrund, Patterns für Novizen eines Fach-



gebiets zu schreiben, kann sich der praktische Wert eines Patterns aus dem individuellen Vermögen und der Gelegenheit zur Umsetzung ergeben.

Dennoch geben nur wenige Pattern-Beschreibungen konkrete Hilfestellungen für eine oder mehrere Umsetzungsmöglichkeiten. Im Software Engineering kann dies beispielsweise durch Code-Beispiele erfolgen (z.B. Gamma et al. [1995]). Im E-Learning oder Interaction Design lassen sich solche Anwendungen als Beispiel anführen, die frei zugänglich und in ihrer Benutzung gut dokumentiert sind. Ungeachtet dessen lässt sich aus einer Musterbeschreibung keine Aussagen über den Realisierungsaufwand treffen. Ressourcen wie Zeit, Geld oder Personal kennzeichnen den Lösungsaufwand, der von der Komplexität der Problemstellung (und ihrer Zerlegbarkeit) sowie den Fähigkeiten der mit der Lösung betrauten Personen abhängt.

**Problemlösung** Wie bereits erwähnt, enthalten Design Patterns die Essenz einer erfolgreichen Lösung für ein wiederkehrendes Gestaltungsproblem. Eine Lösung zeigt die Grenzen des Gestaltungsraums für verschiedene Lösungsvarianten auf, die dem Problem gerecht werden [Kohls 2014, S. 40]. Durch Abstraktion von zurückliegenden Erfahrungen, bereits existierenden Produkten oder ähnlichen Sachverhalten wird versucht ein Mindestmaß an Informationen als Lösung festzuhalten, welches für die Umsetzung ausreicht. Entgegen einer Standardlösung gebietet der Lösungsansatz dem Gestalter genügend Freiheit, die Lösung hinsichtlich seiner aktuellen Anforderungen anzupassen, ohne damit den Kern der Lösung zu verfehlen. Kohls [2014, S. 39] spricht in dem Zusammenhang von strukturerhaltenden Variationen. Entscheidend ist dabei jedoch das richtige Maß an Abstraktion. Das Maß ist von der Zielgruppe bzw. von den Kompetenzen und dem Vorwissen des Betrachters abhängig. Erscheint eine Lösung subjektiv als zu abstrakt, wird es schwer, die Lösung im Sinne des Patterns umzusetzen. Ist eine Lösung hingegen zu konkret und detailbehaftet, schränkt sie den Gestaltungsspielraum und somit die Vielfalt an möglichen Instanzen des Patterns unnötig ein. Kohls [2014] führt drei Möglichkeiten an, etwas zu abstrahieren:

- Isolation von relevanten Merkmalen durch Weglassen nicht relevanter Merkmale,
- Verallgemeinerung von Merkmalen durch Vernachlässigung von Abweichungen,
- Bündelung anhand emergenter Merkmale, während strukturelle Merkmale bei Seite gelassen werden.

Im Interaction Design werden beispielsweise Angaben wie Schriftgrößen, Farben, Größenangaben und -verhältnisse nicht im Pattern beschrieben. Diese Art einer stilistischen Reduktion erfolgt zugunsten funktionaler Aspekte. Pädagogische Pattern abstrahieren vom Unterrichtsfach bzw. dem Lerngegenstand zugunsten übergreifender Merkmale. Software Engineering Patterns abstrahieren (im Idealfall) von der Programmiersprache als strukturelles Merkmal. Ähnlich verhält es sich in der Architektur, wo Baumaterialien in den Patterns außer Acht bleiben können.

Corfman [1998] betont die Auswahl zwischen alternativen Lösungen für ein Problem als Ausdruck der Gestaltungsfreiheit. In ähnlicher Weise liegt auch die Auswahl ergänzender Patterns im Ermessen des Betrachters.

**Konsequenzen der Lösung** In einer Musterbeschreibung werden durch die Konsequenzen Vor- und Nachteile angeführt, die mit der Umsetzung der Lösung verbunden sein können. In diesem Abschnitt werden die zuvor dargelegten Forces aufgegriffen und in Anbetracht der Lösung diskutiert [Wellhausen und Fiesser 2012]. Manche Forces werden durch die Lösung ausgeglichen, während andere bestehen bleiben oder sogar im Konflikt mit der Lösung stehen. Durch die Gegenüberstellung der Vor- und Nachteile lässt sich der Nutzen gegenüber den Kosten der Umsetzung abschätzen. Die Diskussion der Lösung wird durch die Konsequenzen strukturiert und bietet dem Leser somit eine Hilfestellung bei der Entscheidungsfindung und Priorisierung der darüber hinaus notwendigen Patterns.

Neue Herausforderungen oder Probleme, die aus der Umsetzung des Patterns resultieren, werden in der Regel durch Verweise auf andere Patterns berücksichtigt. Die Verweise kennzeichnen somit einen Pfad zu den Patterns, die in Anbetracht der gesamten Gestaltungsaufgabe als nächstes zu berücksichtigen sind. Maßgebend ist dabei immer der Kontext und der Rahmen des individuellen Gestaltungsvorhabens.

Typische positive Konsequenzen im Interaction Design im Bezug auf Videos betreffen:

- die Kenntlichmachung von Details (visuell oder auditiv),
- die Erleichterung des Zugangs und
- die Adaptivität und benutzerdefinierten Anpassungen.

Zu den typischen negativen Konsequenzen zählen:

- mangelnder Platz zur Darstellung von Informationen und Interaktionselementen,
- Komplexität der Benutzerführung (z.B. Anzahl Klicks, Orientierung, Selbstbeschreibungsfähigkeit),
- hohe kognitive Last (z.B. Anzahl oder Dynamik der Informations-/Interaktionselemente).

Im Bereich des Lernens führt Kohls [2014] als typische positive Konsequenzen faires Feedback, Motivation von Studierenden, tiefgreifendes Verständnis des Lerngegenstands, Kreativität sowie die Reduktion der Arbeitslast für Studierende und Lehrende an. Typische negative Konsequenzen umfassen z.B. eine lange Vorbereitungszeit, wechselndes Engagement von Studierenden, Abhängigkeit von Technologie sowie hohe finanzielle Kosten.

**Beispiele oder Instanzen eines Musters** Die in den Musterbeschreibungen enthaltenen Beispiele verschaffen Klarheit und stellen insbesondere für weniger erfahrene Anwender eine Hilfe bei der Umsetzung des Musters dar [Tidwell 2011]. Mit Hilfe von Beispielen lässt sich die Bandbreite möglicher Ausgestaltungen im Sinne von Lösungsvarianten darlegen. Idealerweise eröffnen sie dem Leser einen einfachen und schnellen Zugang zur Lösung. Bildliche Darstellungen können Anwender jedoch dazu verleiten die Beispiele zu sehr als Lösung anzusehen, ohne den Kern der Lösung darin zu erkennen.

Die Art der Repräsentation von Beispielen ist dabei vom Anwendungsfeld abhängig. Von kurzen Auflistungen, bis hin zu umfangreichen Fallstudien variieren die Beispiele in Umfang und Form. Folgende Repräsentationsformen sind gebräuchlich:



- Illustrationen (z.B. „visual sketch“, [Iba 2015]),
- Abbildungen (z.B. Fotos von Gebäuden [Alexander et al. 1977], Unterrichtssituationen [Arvola 2006], Screenshots von Anwendungen [Tidwell 2011]),
- Strukturdiagramme (z.B. UML-Diagramme, [Gamma et al. 1995]),
- Programmcode bzw. Pseudocode (z.B. Gamma et al. [1995]),
- ausführbare Beispiele (z.B. Einbeziehung von ausführbaren Komponenten des *User Interface* in die Beschreibung des Musters),
- Verweise auf Literatur oder Anwendungen im WWW (z.B. Crumlish und Malone [2009]; Yahoo! [2012]).

Neben der Darstellung beispielhafter Lösungen wird in manchen Patterns auch auf Werkzeuge hingewiesen, die bei der Umsetzung der Lösung helfen (z.B. Bauer und Baumgartner [2012]).

**Verwandte Muster** Referenzen heben die Beziehungen zu anderen Mustern hervor und sollen den Anwender dazu anregen, ähnliche oder alternative Muster nachzuschlagen und bei der Lösung einer komplexen Gestaltungsaufgabe einzubeziehen. Man unterscheidet Verweise auf Muster innerhalb oder außerhalb der Mustersprache. In der Musterbeschreibung sind jedoch nur diejenigen Beziehungen in Gestalt von Kapitälchen typografisch hervorgehoben (z.B. USER TRACES), die vom vorliegenden Muster auf ein anderes verweisen. Alle eingehenden Verweise bleiben dem Betrachter in der Regel verborgen. In manchen Beschreibungsformaten werden die Verweise in einer gesonderten Kategorie zusammengefasst (z.B. „Related Patterns“ [Kunert 2009]). Im Vergleich zu gedruckten Pattern-Beschreibungen erleichtern Hyperlinks im Web den Aufruf von Pattern.

#### 4.1.2 Mustersprache

Design Patterns definieren keine isolierten Lösungen [Kunert 2009, S. 67], statt dessen stehen diese in verschiedener Art und Weise miteinander in Beziehung. Beispielsweise werden alternative oder ergänzende sowie über- oder untergeordnete Lösungen referenziert. Eine Pattern Language ist somit eine latente Eigenschaft eines Patterns, da keine Lösung von einer anderen unabhängig ist [Alexander 1979, S. 306]. Da Patterns oft kleinteilige Lösungen für kleinteilige Probleme beschreiben, erscheinen Verweise auch aus praktischen Gründen notwendig, um komplexere Gestaltungsaufgaben durch die Kombination von Patterns lösen zu können. Trotz der Komplexitätsreduktion durch eine kleinteilige Beschreibung von Lösungen und Problemen, muss es schließlich möglich sein, komplexe Systeme zu schaffen. Die Summe der Teile wird in einer Pattern Language nicht auf die Patterns als Einzelteile reduziert, sondern gewinnt durch deren Beziehungen.

Verweise auf andere Patterns sind jedoch auch ein immanenter Bestandteil der Beschreibung eines Patterns, da zum einen die Probleme in Wechselwirkung mit anderen Lösungen stehen und zum anderen sich aus einer Lösung wieder neue Probleme ergeben. Anstatt im Text ein verwandtes Problem mehrfach mit Lösungen in seinem Kontext darzulegen, beschränkt man sich auf den Verweis. Pattern Languages sind insofern auch eine logische Konsequenz des Beschreibungsformats, welches eine Betrachtung in unterschiedlichen Detailstufen ermöglicht. Der Name des Patterns enthält vergleichsweise wenig Informationen, während erst die Lösung

und das Problem, gefolgt vom Kontext die Anwendbarkeit präzisieren. In der nächsten Detailstufe folgen Forces und Konsequenzen. Erst dann eröffnen die Verweise auf Patterns innerhalb der selben Pattern Language eine relativ unkomplizierte Möglichkeit zur Vertiefung, indem sich das Pattern durch die referenzierten Patterns entfaltet. Die nächste Detailstufe besteht aus externen Ressourcen wie z.B. Patterns verwandter Domänen oder Literatur im Allgemeinen.

Buschmann und Henney [2008] weisen in diesem Zusammenhang darauf hin, dass eine Pattern Language sowohl einen Gegenstand, als auch einen Prozess darstellt. Die Querbeziehungen weisen den Weg, wie ein bestimmtes System in einem bestimmten Bereich zu gestalten ist und welche Gestaltungsoptionen für das System in einem Gestaltungsraum bestehen.

Weniger intuitiv verständlich dürfte der Begriff der Sprache im Zusammenhang mit Patterns erscheinen. Der Begriff bezieht sich weder auf natürliche Sprachen, noch auf künstliche Sprachen, wie etwa Programmier- oder Auszeichnungssprachen. Gemeint ist vielmehr eine Formensprache, mit der der Gestaltungsraum in einem Anwendungsgebiet (z.B. der Architektur, Stadtplanung oder objektorientierten Programmierung) beschrieben werden kann. Eine Pattern Language ist somit ein Mittel, um die möglichen Operationen, um einen Artefakt zu gestalten zu kommunizieren. Die Kommunikation beschränkt sich dabei nicht auf den Text der Pattern-Beschreibung, sondern auch auf den Sprachgebrauch von Experten, welche den Namen eines Patterns als Synonym für das damit verbundene Pattern gebrauchen. Beispielsweise wissen Software Entwickler, wie ein OBSERVER zu implementieren ist, während Interaction Designer verschiedene Ausprägungen einer INFINITE LIST kennen.

Ein kurzer Exkurs in die Linguistik zeigt uns eine weitere Erklärung für die Verwendung des Begriffs einer Sprache. Aus linguistischer Sicht besteht die einfachste Form einer Sprache aus einer Menge an Elementen oder Symbolen und einer Menge an Regeln, die diese Symbole verbinden. In einer natürlichen Sprache sind Wörter die Elemente und eine Grammatik legt fest, welche Folge von Wörtern zulässig ist [Alexander 1979, s. 185f]. Eine abzählbare Menge an Wörtern sowie eine Grammatik mit abzählbar vielen Regeln dient uns als ein System zur Bildung unendlich vieler wohlgeformter Wortfolgen und Sätze [Chomsky 1957, S. 14]. In ähnlicher Weise verhält es sich mit Design Patterns und einer Pattern Language, wobei die Patterns Elemente und Regeln zugleich sind. Ein Pattern ist ein Element der Pattern Language und enthält zugleich Verweise, aus denen eine Folge von Patterns entstehen kann. Da nicht unbedingt alle Kombination von Patterns sinnvoll erscheinen, sind explizite Verweise auf „related patterns“ Bestandteil vieler Patterns [Kohls 2014, S. 53].

In der Literatur gibt es eine Vielzahl von Definitionen für Pattern Languages. Im Folgenden ist eine markante Auswahl zu lesen, aus der eine für diese Arbeit gültige Definition abgeleitet wurde:

- „A pattern language is a collection of patterns that build on each other to generate a system.“ [Buschmann und Henney 2008]
- „[A pattern] references higher-level patterns describing the context in which it can be applied, and lower-level patterns that could be used after the current one to further refine the solution. This hierarchy structures a comprehensive collection of patterns into a pattern language.“ [Borchers 2000]

- „Pattern languages have three essential elements. First, the language has to contain a standard pattern definition. [...] Secondly, the language must logically group patterns. [...] Thirdly, pattern interrelationships should be described.“ [Seffah 2010]

**Definition 5.** *Eine Pattern Language ist ein System aus Patterns, die durch hierarchische oder netzartige Verweise miteinander in Beziehung stehen.*

**Struktur und Klassifikation** Eine Strukturierung und Klassifikation von Mustern innerhalb einer Sammlung oder Sprache soll den Anwendern derselben bei der Auswahl und Anwendung der Muster unterstützen. Im Sinne der besseren Benutzbarkeit werden die Strukturen in Listen [Alexander et al. 1977], gerichteten Graphen [Gamma et al. 1995] oder in UML-Diagrammen [Derntl und Motschnig-Pitrik 2004] visualisiert. Die Struktur einer Mustersprache ergibt sich aus dem Umstand, dass individuelle Muster keine isolierten Lösungen darstellen [Alexander 1979, S. 311]. Durch explizite Verweise in Gestalt der Forces entsteht ein Netzwerk von Verbindungen zwischen den einzelnen Mustern [Alexander 1979, S. 305]. Christopher Alexanders Mustersprache besitzt neben dieser netzartigen Struktur eine hierarchische Ordnung [Alexander 1979]. Zum einen enthält ein Muster untergeordnete Muster, während es selbst wieder Teil eines übergeordneten Konzepts ist [Alexander 1979, S. 312]. Diese Beziehung zwischen detaillierten und mehr abstrakten Mustern folgt dem Vererbungsprinzip, indem das detaillierte Muster vom abstrakten bestimmte Abschnitte erbt [Derntl und Motschnig-Pitrik 2004].

Zum anderen skalieren die Muster innerhalb der Mustersprache, wie man am Beispiel der alexandrinischen Patterns sehen kann: Angefangen bei physikalischen Materialien und Strukturen bei der Konstruktion (z.B. COLUMN CONNECTIONS, SITTING WALL), über die Form und das Leben in Gebäuden (z.B. SLEEPING TO THE EAST, FLEXIBLE OFFICE SPACE) bis hin zu Strukturen der Stadtplanung (z.B. FOUR-STORY LIMIT, ACCESS TO WATER) [Alexander 1979, S. 306]. Die alexandrinische Ordnung der Muster resultiert aus dem Gegenstand der Sprache: Der dreidimensionalen Welt der vom Menschen geschaffenen Architektur.

Im Gegensatz zum geometrisch Begreifbaren entwickelten Gamma et al. [1995] die Struktur ihrer Musterkollektion anhand von Datenstrukturen. Dabei werden Muster anhand ihrer Gültigkeitsbereiche in klassen- oder objektbasierte Muster unterteilt. Erstere ergeben sich aus der Vererbung von Klassen und sind damit „statisch, das heißt zur Übersetzungszeit festgelegt“ [Gamma et al. 1995, S. 14]. Letztere sind Beziehungen zwischen Objekten. Diese sind dynamisch und können sich während der Laufzeit ändern [Gamma et al. 1995, S. 14]. In einer weiteren Dimension unterscheiden Gamma et al. [1995] Muster hinsichtlich drei verschiedener Funktionen im Software Programm:

- **Erzeugende Muster:** beschreiben Prozesse der Objekterzeugung (z.B. FACTORY METHOD),
- **Strukturorientierte Muster:** befassen sich mit der Zusammensetzung von Klassen und Objekten (z.B. DECORATOR),
- **Verhaltensorientierte Muster:** charakterisieren Art und Weise, in der Klassen und Objekte zusammenarbeiten und Zuständigkeiten aufteilen (z.B. MODEL-VIEW-CONTROLLER).

Schümmer und Lukosch [2007] gliedern ihre Patterns für die computervermittelte Kommunikation auf drei Ebenen: (1) *Community Support*, (2) *Group Support*, (3) *Base Technology*. Während die ersten beiden Ebenen die Anzahl der beteiligten Personen berücksichtigen, ist die dritte

Ebene, ähnlich wie bei Alexander, technischen Problemen und Problemlösungen vorbehalten. Auf jeder Ebenen bestehen zudem thematische Cluster.

Im Kontext des interaktiven Fernsehens (iTV) entwickelte Kunert [2009, S. 185ff, 225f] ebenfalls eine hierarchische Ordnung von Design Patterns, die sich jedoch an einer von ihm ermittelten Hierarchie von Gestaltungsproblemen<sup>16</sup> aus Sicht des Designers orientiert. Die für Software Entwickler und UI-Designer konzipierten Mustersammlungen von Derntl und Motschnig-Pitrik [2004], Tidwell [2005] sowie Crumlish und Malone [2009] sind thematisch gemäß einer Hierarchie von Gestaltungsproblemen geordnet, mit denen sich die Zielgruppe konfrontiert sieht. Besonders deutlich wird das bei van Welie und Veer [2003], die für Webdesign Patterns zunächst den Typus von Seite („posture type patterns“, wie z.B. Webseiten von Kleinunternehmen), dann das Ziel der Seite („experience patterns“, wie z.B. Shopping) und dann Patterns auf der Ebene von konkreten Aufgaben definieren, die sich aus Patterns für Aktionen zusammensetzen.

Einen ganz anderen Weg haben Santos et al. [2011] eingeschlagen, indem sie Mustersprachen für verschiedene Zielgruppen – wie Lehrende, Entwickler, Spezifikationsexperten – miteinander verknüpfen. Das *Problem Statement* eines Musters in einer Mustersprache wird dabei zusätzlich mit einer Lösung eines anderen Patterns adressiert, welches zu einer anderen Mustersprache gehört. Am Beispiel der Entwicklung von QTI-kompatiblen E-Assessment-Aufgaben auf Basis von Google-Karten ermöglichen sie Mustersprachen über Fachgrenzen hinweg.

### 4.1.3 Verwandte Ansätze im Interaction Design

Neben Entwurfsmustern existieren eine Reihe weiterer Ansätze, um gute Lösungen und ihre zugrunde liegenden Prinzipien sowie das implizite Wissen von Designern zu dokumentieren. Noch bevor Design Patterns im Bereich der Mensch Computer Interaktion 1998 durch Jennifer Tidwell eingeführt wurden [Tidwell 1998], fanden *Design Principles*, *Guidelines* und *Style Guides* bei der Konzeption, Gestaltung, Realisierung und Evaluation von *User Interfaces* eine breite Anwendung. *Guidelines* spielen heute jedoch nur noch eine untergeordnete Rolle [Seffah und Taleb 2012]. *Principles* wurden nicht widerlegt, jedoch aufgrund ihrer Abstraktheit auch nicht angewendet. *Style Guides* hingegen gelten immer noch als wichtige Dokumente zur Festlegung einheitlicher Gestaltungsrichtlinien innerhalb von Organisationen. Darüber hinaus gibt es noch weitere Formate zur Manifestation und Kommunikation von Handlungsempfehlungen (z.B. *Claims* [Seffah und Taleb 2012]), die sich jedoch nicht durchgesetzt haben. Die einzelnen Beschreibungsformen sind im einzelnen nicht immer eindeutig zu differenzieren. Es existieren Mischformen (z.B. Apple Computers Inc. [1992] oder Költringer et al. [2005]). Im Folgenden werden die einzelnen Formate genauer vorgestellt, um sie deutlich von Design Patterns abgrenzen zu können. Ausgenommen sind hiervon Alternativen zu didaktischen Design Patterns, welche jedoch im darauffolgenden Kapitel in Bezug zu CSCL-Scripts kurz behandelt werden.

---

<sup>16</sup>Die Hierarchie von Gestaltungsproblemen in Bezug auf iTV-Applikationen resultiert aus Analysen des Anwendungskontext, anderen Gestaltungsrichtlinien, Anforderungen von Designern, Benutzeraufgaben und -anforderungen.

**Design Prinzipien (*Design Principles*)** Im Bereich der HCI konstatieren Design Prinzipien weitreichende Ziele hinsichtlich der Usability – sprich dem Grade, in dem ein System vom Menschen bedienbar sein soll. Sie gelten als eine Art von Gestaltungsrichtlinie mit dem höchsten Abstraktionsgrad. Aus diesem Grund sind sie auf sehr viele Kontexte anwendbar. Design Prinzipien werden insbesondere bei der Konzeption und Evaluation interaktiver Anwendungen angewendet. Sie basieren entweder auf psychologischen Theorien oder auf praktischen Erfahrungen. Zu den bekannteren Design Prinzipien zählen Ben Shneiderman's „Eight golden rules of user interface design“ [Shneiderman und Plaisant 2004, S. 74], die „Ten Usability Principles“ von Nielsen und Molich [1990] sowie die Grundsätze der Dialoggestaltung in Form der Richtlinie DIN 9241-110 [Deutsches Institut für Normung (Hsg.) 2003]. Ein Überblick der verschiedenen Prinzipien findet sich bei Dumas und Redish [1999].

Vergleichbar sind in diesem Zusammenhang auch die aus den kognitiven Theorien abgeleiteten und im Abschnitt 2.2 (S. 36) besprochenen Prinzipien des Lernens mit Multimedia [Mayer 2009] bzw. jene zur Reduzierung der kognitiven Belastung [Clark et al. 2006]. Diese Prinzipien können helfen, die Verarbeitung von auditiven und visuellen Informationen bei der Gestaltung von Lernressourcen zu verbessern, wobei die Interaktion mit dem System sowie die Interaktion zwischen den beteiligten Personen dabei außer Acht gelassen wird.

Aufgrund ihrer abstrakten Form zeigen *Design Principles*, so Kunert [2009, S. 63], keinen konkreten Weg zur Umsetzung auf. Stattdessen lassen sie viel Raum für Interpretation, Präzisierung und Erweiterung hinsichtlich der Anwendung in einem spezifischen Kontext. Insbesondere weniger erfahrene UI-Designer oder -Entwickler erhalten dadurch nicht ausreichend Unterstützung [Tidwell 2011].

**User Interface Guidelines** Im Vergleich zu Design Prinzipien beinhalten *Guidelines* wesentlich konkretere Anweisungen auf einem weniger abstrakten Niveau. Während Design Prinzipien die Gestaltungsziele definieren, beschreiben Gestaltungsrichtlinien den Weg zur Erreichung derselben im jeweiligen Anwendungskontext [Kunert 2009, S. 64f].

Der Inhalt von Guidelines ist sehr unterschiedlich. Neben allgemeinen UI-Gestaltungsprinzipien, existieren auch Richtlinien für Anwendungen auf spezifischen Plattformen [Henninger et al. 1995; Tetzlaff und Schwartz 1991].

Einige dieser Richtlinien, wie etwa die ISO 9241 [Deutsches Institut für Normung (Hsg.) 2003] oder ISO 14915, sind ebenfalls als internationale Standards verbrieft. Sie beruhen oft nur auf Empfehlungen von Experten mit entsprechender Autorität und Erfahrung in dem jeweiligen Anwendungsgebiet.

Auffällig sind jedoch die strukturellen Eigenschaften der Beschreibungsformen, wie etwa die Guidelines für DVD-Menüs von Kappel et al. [2006] bzw. Költringer et al. [2005]. Darin finden sich typische Strukturelemente von Patterns wieder, wobei die Lösungen ohne eine Beschreibung des ursächlichen Problems hervortreten. Der Kontext als dritter Kernbestandteil eines Pattern ist entweder überhaupt nicht definiert, d.h. vollkommen offen oder aber implizit durch Bezug auf ein ganz konkretes Anwendungsgebiet, eine eingeschränkte Gruppe von Nutzern oder ein einzelnes technisches System bzw. einen Standard dargelegt. Aus Sicht eines Desi-

gners ist es schwierig die Anwendbarkeit einer Lösung einzuschätzen, da selbige nicht anhand der resultierenden Konflikte und Wechselwirkungen diskutiert wird.

Henninger et al. [1995] kritisieren die fehlenden Bezüge zur Lösung spezifischer Aufgaben anhand der vordergründig beschriebenen Komponenten. In einer kleinen Benutzerstudie haben Tetzlaff und Schwartz [1991] die Bedeutung und praktische Anwendung von Guidelines untersucht. Im Ergebnis zeigten sich Schwierigkeiten, die Guidelines zu verstehen und richtig anzuwenden. Insbesondere konzentrierten sich die Teilnehmer dieser Untersuchung mehr auf die illustrierten Beispiele und realisierten auch unwesentliche Details derselben.

**Style Guides** Als die detaillierteste Form der hier vorgestellten Richtlinien für die Gestaltung von Benutzerschnittstellen gelten die *Style Guides*. Als *Style Guides* bezeichnet man im User Interaction Design Empfehlungen für Software Anbieter mit dem Ziel, Benutzeroberflächen in einer möglichst einheitlichen Weise zu gestalten. Typische Bestandteile sind Angaben zu Farbpaletten, Typografie, Layout, Logos und Piktogrammen sowie Menüstrukturen und Navigationselementen. Die erste Anleitung dieser Art wurde bereits 1992 unter dem Titel „Macintosh Human Interface Guidelines“ [Apple Computers Inc. 1992] veröffentlicht. Für die unmittelbare Anwendung in der Gestaltung bestimmt, lassen sie kaum Raum für Interpretationen. Demzufolge bedingt der hohe Detailgehalt ihren geringen Abstraktionsgrad. *Style Guides* lassen sich nur schwer auf andere Anwendungen und Anwendungskontexte, z.B. von Desktop- auf Web- oder auf Mobilanwendungen, übertragen [Kunert 2009, S. 65f]. Bedingt durch technische Neuerungen und weiterführende Interaktionsmöglichkeiten veralten diese Lösungen relativ schnell.

Zusammenfassend werden in Tab. 4.1 die drei hier vorgestellten Gestaltungsrichtlinien mit Interaction Design Patterns verglichen. Patterns zeichnen sich vor allem dadurch aus, dass sie die Anwender im Gestaltungsprozess unterstützen, anstatt lediglich abstrakte Ziele vorzugeben. Aus diesem Grund sind Pattern Languages vergleichsweise umfangreiche Werke von oftmals mehr als 100 Textseiten. Sie stellen jedoch einen Kompromiss zwischen sehr speziellen und kaum übertragbaren *Style Guides* und den sehr abstrakten *Design Principles* sowie *Guidelines* dar. Die in Patterns zum Ausdruck kommende Essenz einer Lösung lässt viel Gestaltungsspielraum für nutzerspezifische Anpassungen, so dass eine Übertragung in verschiedene Anwendungskontexte möglich wird (z.B. vom Desktop auf Web- oder Mobilanwendungen). Der wesentliche Unterschied besteht jedoch darin, empirische Belege in Form von bekannten, funktionierenden Lösungen vorweisen zu können, anstatt sich allein auf die Erfahrungen weniger Experten (Design Principles) zu berufen.

### 4.1.4 Verwandte Pattern Languages

Nach der Betrachtung verwandter Ansätze zur Vermittlung von Wissen über die Gestaltung von Benutzerschnittstellen im User Interaction Design richtet sich das Augenmerk nun auf bestehende Patterns und Pattern Languages. In Anbetracht der Forschungsfrage FF 1 stehen Patterns der Bereiche Interaction Design, Mensch-Computer-Interaktion, Multimedia (inkl. Videos), Hypermedia im weiteren Sinne und E-Learning Patterns im Fokus.

Im Interaction Design gibt es eine Reihe von Patterns und Pattern Languages. Jenifer Tidwell gilt als Pionierin der Interaction Design Patterns. Nach dem Debut [Tidwell 1998] folgte ein



**Tabelle 4.1:** Vergleichende Darstellungen verschiedener Gestaltungsrichtlinien

Richtlinie	Abstraktionsgrad	Übertragbarkeit	Orientierung	Empirische Belege	Textumfang
Design Principles	hoch	hoch	zielorientiert	Erfahrungen, teilweise durch Studien belegt	gering
Guidelines	hoch	hoch	teilw. prozessorientiert	Erfahrungen	mittel
Style Guides	gering	gering	zielorientiert	–	mittel
Design Patterns	mittel	mittel	prozessorient	Beispielinstanzen, teilw. Studien	hoch

umfassende Sammlung mit Patterns mit generischen Interaktionselementen für Desktop- und Web-Anwendungen [Tidwell 2005], die 2011 um weitere Patterns, u.a. für Mobilgeräte ergänzt wurde [Tidwell 2011]. Tidwells Patterns sind jedoch generischer Natur, ohne Schwerpunkte auf bestimmte Anwendungsbereiche zu legen. Demzufolge finden sich darin beispielsweise keine Patterns, die spezifische Probleme zeitabhängiger Medien oder Videoanwendungen im Allgemeinen adressieren. Van Duyne et al. sowie van Welie und Veer haben spezifischere Design Patterns für Web-Anwendungen vorgelegt, die im Wesentlichen typische Hypertext- und Bedienelemente im Kontext des Web 2.0 enthalten, ohne dabei gesondert auf Videoanwendungen einzugehen [van Duyne et al. 2006; van Welie und Veer 2003]. Darüber hinaus wurden einige wenige Sammlungen von Patterns im Umfeld von Unternehmen (z.B. Yahoo! [2012] oder Infragistics [2013]) erarbeitet. Letztere weisen viele Übereinstimmungen mit den Patterns der zuvor genannten Autoren auf.

In thematisch ähnlichen Domänen wie dem interaktiven Fernsehen, zeitbasierter *Hypermedia* oder bei der Produktion von *Screencasts* geben *Interaction Design Patterns* als Beschreibung guter Gestaltungslösungen eine Orientierung für die Konzeption und Entwicklung entsprechender Systeme [Chen und Rabb 2010; Kunert 2009; Lopes und Carriico 2007]. Mediendidaktische Aspekte wurden in letztgenannten Arbeiten, wenn überhaupt, nur marginal berücksichtigt. Videolernumgebungen unterscheiden sich durch ihren Anwendungskontext und Ausführungsumgebung, wie aus den Kapiteln 2 und 3 hervorgeht, deutlich von technischen Systemen im Bereich des Entertainments oder (Internet-)Fernsehens.

Auch im Bereich des E-Learnings gibt es bereits einige Vorarbeiten, die auf Patterns aufbauen und auf diese Weise erfolgreiche Lösungen und Praktiken des computer- und internetgestützten Lernens und Lehrens vermitteln (vgl. Überblick bei Kohls [2014]). Insbesondere das von Blessing und Kortenkamp vorgelegte Pattern VIDEOCLIPQUEST [Blessing und Kortenkamp 2009, 2010; Kortenkamp und Blessing 2011] weist in die hier verfolgte Richtung, Patterns für Videolernumgebungen zu ermitteln. VIDEOCLIPQUEST integriert Überprüfungsaufgaben in ein Video, wobei Frage und Antwort an unterschiedlichen zeitlichen Positionen verankert sind. Die didaktische Intention dabei beschreiben die Autoren mit dem Begriff der „Aufgabenmaskierung“ („mask the task“). Eine Frage wird demnach so vage formuliert, dass erst durch die Betrachtung der nachfolgenden Szene im Video der Kontext vermittelt wird. Auf die Frage „Was war der Auslöser für diesen Konflikt“, kann der Lernende erst dann eine Antwort geben, wenn



sich ihm der Kontext, also der Konflikt selbst, darstellt. Auf diese Weise sollen Lernende dazu angeregt werden, sich ein Video anzusehen, um anschließend die Aufgabenlösung einzugeben. Abgesehen von diesem Einzelfall, finden sich in E-Learning-Patterns nur sehr wenige Bezüge zum Lernen mit Videos und dem Lernen innerhalb von Videolernumgebungen. Videos fungieren als atomare, d.h. unteilbare Bausteine in Kursen. Nichtsdestotrotz sind folgende Pattern Languages erwähnenswert:

- Avgeriou et al. [2003a,b] stellen Patterns für *Learning Management Systems* und gehen dabei auf typische Komponenten zur Kursverwaltung, Erstellung von Inhalten und Gruppenarbeit ein.
- Bauer und Baumgartner [2012] stellt in einer Sammlung von Patterns für E-Portfolios vor, in denen das individuelle, reflektierte und kollaborative Lernen sowie die Organisation und der Ablauf der Lehre mit Hilfe von E-Portfolios im Mittelpunkt steht.
- Chen und Rabb [2010] erarbeiteten Patterns für die Produktion von Screencasts und gehen auf die Präsentation, Fokussierung und Strukturierung der Inhalte ein.
- Goodyear et al. [2006] fassen theorie- und erfahrungsgestützte Patterns zusammen, die dem *Networked Learning* im Sinne von kollaborativen und kooperativen Lernen mit Hilfe von Informations- und Kommunikationstechnologien gewidmet sind. Sie kategorisieren die Patterns hinsichtlich der Aktivitäten der Gruppe und des Lehrenden sowie hinsichtlich der Phasen des Verlaufs einer E-Learning-Veranstaltung („start-up, beginning, middle, end“), für die sie jeweils unterschiedliche Zwecke bestimmen.
- Holden et al. [2010] haben acht Patterns für die gemeinsame Konzeption und Entwicklung von Lerninhalten und -aktivitäten identifiziert.
- Kohls [2014, S. 303] stellte eine kleine Sammlung von Patterns zur Vorbereitung und Durchführung von Online Trainings bzw. Webinaren vor, die bei *e-teaching.org* mit Hilfe der Videokonferenzsoftware *Adobe Connect* durchgeführt wurden.
- Köhne [2005] entwickelte in ihrer Dissertation Patterns für die Konzeption, Organisation und Durchführung von Blended Learning Veranstaltungen und evaluierte diese im praktischen Einsatz.
- Schadowitz [2009] widmet der kulturübergreifenden, computergestützten Kollaboration mehrere Patterns und prototypische Patterns. Darin kommen Aspekte der Gruppenkoordination, Kontextualisierung der Kommunikation sowie technische Belange hinsichtlich der gemeinsamen Bearbeitung von Inhalten zum Ausdruck.
- Schümmer und Lukosch [2007] haben eine umfangreiche Sammlung von Patterns für die computervermittelte Interaktion in Anwendungen, die von mehr als nur einem Anwender (synchron oder asynchron) genutzt werden,
- Salah und Zeid [2009] haben Patterns für die Konzeption und Entwicklung intelligenter tutorieller Systeme geschaffen.

Zusammenfassend ist festzustellen, dass es bislang keine Leitfäden, Gestaltungsprinzipien oder Design Patterns für die Konzeption und Gestaltung von Videolernumgebungen gibt. Der Bedarf für derartige Hilfestellungen ergibt sich aus den komplexen Anforderungen an Videolernumgebungen, die sich nicht nur aus technologischen, sondern auch aus didaktischen und kognitionspsychologischen Aspekten ergeben. Der durch Patterns repräsentierte Ansatz Lösun-

gen für komplexe Probleme durch ein kleinteiliges Netz aus Teilproblemen und Teillösungen im jeweiligen Kontext zu beschreiben, ist deshalb sehr vielversprechend. Der wesentliche Vorteil besteht darin, dem Anwender einen angemessenen Gestaltungsspielraum einzuräumen, indem erstens individuelle Anforderungen durch eine selbstbestimmte Auswahl an Patterns adressiert werden können, zweitens ein Pattern lediglich die Essenz einer Lösung beschreibt, sodass dennoch individualisierte und in ihrem Design einzigartige Anwendungen geschaffen werden können und drittens der Anwender auf die Bandbreite möglicher Gestaltungsformen innerhalb des spezifischen Gestaltungsraums von Hypervideos hingewiesen wird. Interaction Design Patterns für Videolernumgebungen bilden somit die Schnittmenge aus guten *User Interface* Lösungen für Hypervideosysteme und E-Learning Anwendungen.

## Systematische Elaboration von Design Patterns

In der Forschungsfrage FF 1 wurde die Frage gestellt, welche wiederverwendbaren, abstrakten Lösungen für häufig auftretende Probleme hinsichtlich der Gestaltung des *User Interfaces* von Videolernumgebungen existieren. Gesucht sind demzufolge *Interaction Design Patterns* für Videolernumgebungen. Zur Beantwortung dieser Frage, bedarf es einer systematischen Vorgehensweise, um Regelmäßigkeiten im *User Interface* von Videolernumgebungen zu identifizieren und in eine Beschreibung in Form von Patterns zu überführen. Die Suche und Identifizierung eines Patterns bezeichnet man als *Pattern Mining*. Diese Metapher betont die Suche nach Erkenntnissen in existierenden Strukturen und Formen von Artefakten. Die betrachteten Artefakte sind wiederum das Ergebnis der Entscheidungsfindung ihrer Erschaffer (z.B. Designer, Entwickler) und insofern auch eine Repräsentation von implizitem Wissen. Beim *Pattern Mining* gilt es insbesondere die wesentlichen von den unwesentlichen Merkmalen zu trennen [Kohls 2014].

An das *Mining* schließt sich der iterative und durch externes Feedback geprägte Schreibprozess (*Pattern Writing*) an. Teil dessen ist auch die Herstellung von Querbeziehungen zwischen den einzelnen Patterns, sodass im Ergebnis eine in sich verknüpfte *Pattern Language* entsteht.

Ausgehend vom Stand der Forschung wird in diesem Abschnitt eine im Bereich des *Interaction Designs* umfassende Methode des *Pattern Minings* vorgestellt und ausführlich beschrieben. Die Methode umfasst vier Schritte:

- *Pattern Mining*:
  - 1) Inhaltsanalyse von Videolernumgebungen
  - 2) Analyse und Integration ähnlicher Muster bzw. Mustersprachen
- *Pattern Writing*:
  - 3) Verfassen sowie Revision der Muster (inkl. *Shepherding* und *Writers' Workshops*)
  - 4) Konstruktion und Revision der Pattern Language

### 4.2.1 Stand der Forschung bzgl. dem Pattern Mining

Zur Identifikation invarianter Merkmale, die gutes von schlechtem Design unterscheiden führt Kohls [2014] in Anlehnung an Alexander [1979] drei Methoden an:

- **Induktiv:** Induktive Ableitung durch Analyse und Beobachtung existierender Fallbeispiele oder praktischen Erfahrungen
- **Induktiv-deduktiv:** Induktive Analyse von negativen Fallbeispielen, aus denen anschließend funktionierende Lösungen abgeleitet werden.
- **Deduktiv:** Ableitung guter Lösungen durch argumentative Begründungen und theoretische Annahmen.

Die *top-down* Entwicklung von Design Patterns durch deduktives Schließen wird von der Pattern Community im Allgemeinen abgelehnt. Der Ansatz ist per se nicht abwegig, sofern ein auf diese Weise identifiziertes Design Pattern auch einer mehrfachen Erprobung im Feld standhält und von anderen Produzenten (z.B. Software Entwicklern) aufgegriffen wird. Köhne [2005] hat auf diese Weise „Educational Patterns“ für das *Blended Learning* erstellt.

Das induktive Vorgehen ist ein *bottom-up* Verfahren, was auf empirischen Belegen von realen Anwendungen beruht, die den Kern des Gestaltungsprinzips demonstrieren. Es bedarf mindestens drei Beispielinstanzen („Rule of Three“ [Fincher 1999; Kohls und Uttecht 2009; Seffah 2010]), die in einem absichtsvollen Schaffensprozess und nicht etwa zufällig oder durch Reproduktion entstanden sind [Fincher 1999]. Da Muster als verbreitete Lösung wiederkehrender Gestaltungsprobleme definiert sind und ohnehin Beispiele aus der Praxis einen Teil der Musterbeschreibung bilden, liegt induktives Vorgehen nahe. Die Autoren von Design Patterns gehen dabei jedoch sehr unterschiedlich vor und kombinieren mehrere Methoden miteinander. Im Folgenden werden exemplarische Ansätze kurz erläutert und abschließend in Tab. 4.2 zusammengefasst. Die hier dargestellten Ansätze erheben keinen Anspruch auf Vollständigkeit, da einige Pattern-Autoren die Vorgehensweise bei der Identifikation und Erstellung ihrer Patterns nicht offenlegen. Letzteres betrifft z.B. Tidwell [2011], Tidwell [2005], Yahoo! [2012], Lopes und Carriico [2007], Cybulski und Linden [1998], van Duyne et al. [2006], Rüping [2009].

Kunert [2009, S. 88-97] verfolgte einen empirischen Ansatz, um Anforderungen an und Funktionalitäten von iTV-Anwendungen zu ermitteln. Er versammelte für seine Untersuchungen 54 Nutzer von iTV-Systemen sowie reguläre TV-Nutzer in acht Fokusgruppen. Aus den Ergebnissen leitete Kunert zum einen inhaltsunabhängige Anforderungen und Funktionen ab (z.B. Suchen, Partizipieren, Kommunizieren, Texteingabe, Hilfe) und zum anderen inhaltspezifische Anforderungen hinsichtlich iTV-Anwendungen für Nachrichtenbeiträge, Dokumentationen, Musik, Unterhaltung, Quiz-Sendungen sowie Anwendungen für Kinder. Anschließend erfolgte ein Abgleich der ermittelten Anforderungen und Funktionen mit belegbaren Gestaltungslösungen. Diese konkreten Lösungen standen für die Untersuchung anhand einer überschaubaren Anzahl von Testsystemen zur Verfügung und konnten deshalb in Usability-Tests evaluiert und bewertet werden [Kunert 2009, S. 100-103].

In Bezug auf E-Learning Systeme schlagen Retalis et al. [2006] eine Triangulation mehrerer Methoden vor, um ein ganzheitliches und situatives Bild zu erzeugen. Folgende Methoden kamen zur Anwendung:

1. Analyse von E-Learning-Systemen hinsichtlich ihrer Funktionalitäten bzw. „features“ und Kategorisierung der Funktionalitäten;
2. Literaturgestützte Entwicklung von Lehr-Lern-Strategien, um Anforderungen für die systemseitige Umsetzung zu ermitteln;
3. Interviews mit Experten auf dem Gebiet der Didaktik und Systementwicklung;
4. Beobachtung von Anwendern bei der Erledigung von Aufgaben;
5. Analyse von Logdaten aus E-Learning-Systemen;
6. Untersuchung von Patterns aus verwandten Wissensgebieten, in dem Fall HCI.

Zur Identifikation von wiederkehrenden Lösungen bei der Gestaltung von Gesellschaftsspielen analysierten Björk et al. [2003] einerseits die verfügbaren Resultate vergangener Gestaltungsprozesse und interviewten andererseits Designer, die sich professionell mit der Spieleentwicklung auseinandersetzen. In einem ersten Schritt der Analyse wurden wiederkehrende Regelsysteme („game mechanics“), Ideen oder Interaktionen als Pattern-Kandidaten zusammengefasst. Als Zweites beschrieben sie die Einzellösungen je Kandidat und betrachteten mögliche Konsequenzen, die aus der Eliminierung der Lösungen entstehen würden. Schritt drei umfasste eine prototypische Umsetzung der im Pattern enthaltenen Lösung. Abschließend wurde das Muster vollständig gemäß dem gewählten Beschreibungsformat formuliert. Im Laufe dieses Prozesses wurden die Schritte für 200 Pattern-Kandidaten durchlaufen, teilweise in veränderter Reihenfolge. Zusätzlich wurden neun *Game Designer* verschiedener Genres interviewt, wobei diese einige Begrifflichkeiten teilten und am Finden struktureller Zusammenhänge – gleich einer Pattern Language – interessiert waren.

Zur Identifikation von „Interaction Design Patterns for Computers in Sociable Use“ untersuchte Arvola [2006] die Nutzung von Unterstützungssystemen, wie sie beispielsweise Bankangestellte bei Kundenberatungen nutzen. Die Ergebnisse gründen auf videografierten Beobachtungen von 49 Studienteilnehmern, teilstrukturierten Interviews sowie 17 halbtägigen Workshops. Anhand der transkribierten Beobachtungsdaten wurde eine hohe Zahl an Analysekategorien (Nutzungsqualitäten) identifiziert und anschließend gruppiert.

Tab. 4.2 gibt einen Überblick über sechs verschiedene Ansätze zu Identifikation von Design Patterns. Diese Aufstellung ist an die Vorarbeiten von Rising [1999], DeLano [1998], und Kohls [2014] angelehnt und stellt darüber hinaus dar, inwieweit die einzelnen Ansätze im Rahmen dieser Arbeit für das Pattern Mining von Interaction Design Patterns von Videolernumgebungen eine Rolle spielen. Vier der sechs Ansätze kamen aufgrund der verfügbaren Datengrundlage bzw. Personen in Betracht. Interviews sowie Pattern Mining Workshops waren nicht realisierbar, da zu Beginn der Untersuchung im Jahre 2010 keine ausreichende Anzahl an Experten identifiziert und gewonnen werden konnte, die über umfangreiche Erfahrungen mit interaktiven Videos im Rahmen des E-Learnings verfügen. Es spricht jedoch grundsätzlich nichts dagegen, Experten in das Pattern Mining einzubeziehen. Anstelle von Einzel- oder Fokussgruppeninterviews, wurden jedoch Meinungen und Hinweise im Zuge der Revision der Patterns beim *Shepherding* und in den *Writers' Workshops* aufgegriffen.

**Tabelle 4.2:** Übersicht grundlegender Methoden des *Pattern Minings*, einschließlich einer kurzen Erläuterung der Anwendungsmöglichkeiten im Rahmen dieser Arbeit.

Methode	Beschreibung	Anwendung im Kontext dieser Arbeit
„Artifactual approach“	Beobachtung und Analyse von Projekten; Gegenüberstellung und Vergleich ähnlicher Systeme (vgl. [Baggetun et al. 2004])	Inhaltsanalyse relevanter Videolernumgebungen
„Mining by borrowing“	Auswertung von Aufzeichnungen (z.B. Artikel, Bücher, Videos), in denen Experten ihre Erfahrungen wiedergeben	Analyse ähnlicher Pattern Languages und Integration/Referenzierung relevanter Patterns
Retrospektion	Selbstbeobachtung und -analyse eigener Arbeiten und Projekte (vgl. [Baggetun et al. 2004])	Entwicklung und Erprobung eigener Videolernumgebungen, siehe dazu Abschnitt 6.3 (S. 246)
Ethnografische Beobachtungen	Beobachtung von Menschen, die einen Artefakt (z.B. eine Software) nutzen und damit arbeiten (vgl. [Baggetun et al. 2004; Martin et al. 2001; Retalis et al. 2006] und [Schümmer 2005, S. 85-91])	Indirekt durch die Erhebung von Logdaten zur Analyse des Nutzerverhaltens in Videolernumgebungen (vgl. [Seidel 2013, 2014b])
Interviews	Befragung von Experten hinsichtlich ihrer eigenen Erfahrungen und selbst identifizierten Patterns (vgl. [Arvola 2006; Björk et al. 2003])	–
Pattern Mining Workshops	Erfahrungen von Experten werden in Fokussgruppen gesammelt, geordnet und zusammengefasst	–

### 4.2.2 Inhaltsanalyse von Videolernumgebungen

Der erste Schritt des *Pattern Minings* besteht aus einer empirischen Inhaltsanalyse nach Kromrey [2011]. Die empirische Inhaltsanalyse definiert Kromrey als:

„eine Forschungstechnik, mit der man aus jeder Art von Bedeutungsträgern durch systematische und objektive Identifizierung ihrer Elemente Schlüsse ziehen kann, die über das einzelne analysierte Dokument hinaus verallgemeinerbar sein sollen.“ [Kromrey 2011, S. 300].

Da die Methode nicht auf eine bestimmte Art von Dokumenten beschränkt ist, lässt sie sich auch auf Software Anwendungen im Allgemeinen und Videolernumgebungen im Speziellen anwenden.

In Bezug auf die Forschungsfrage FF 1 besteht das Ziel der Analyse darin, den Stand des *User Interface Designs* von Videolernumgebungen systematisch und nachvollziehbar zu erfassen. Insbesondere gilt es, die Verbreitung von bestimmten Funktionalitäten zu erheben. Die Auftrittshäufigkeit dient dabei als Beleg für die Relevanz, welche Designer und Entwickler einem bestimmten Merkmal beigemessen haben. Um so größer die Anzahl an Lernumgebungen mit einem bestimmten Merkmal, desto relevanter ist dieses Merkmal für die Ausgestaltung neuer Videolernumgebungen mit Hilfe eines entsprechenden Design Patterns. Die Betrachtung möglichst vieler Beispielinstanzen soll einerseits dazu beitragen, den Kern der dahinter liegenden Lösungen zu bestimmen und andererseits die Bandbreite an Variationsmöglichkeiten zu erschließen.

In diesem Teilabschnitt soll zunächst die Grundgesamtheit der untersuchten Informationsträger erläutert werden. Im zweiten Teilabschnitt geht es um die Bestimmung der Kategorien, die man den einzelnen Informationsträgern in der Analyse zuschreibt. Abschließend wird die Vorgehensweise beschrieben, mit der 121 Videolernumgebungen ausgewertet wurden.

**Bestimmung der Grundgesamtheit** Die Grundgesamtheit der Informationsträger setzt sich aus Anwendungen zusammen, die das Lernen mit Videos ermöglichen. Um eine möglichst große Bandbreite an Videolernumgebungen untersuchen zu können, unterscheiden sich die Informationsträger hinsichtlich ihrer Domäne (Wissenschaft, Kunst, Bildung, Unterhaltung), ihrer Zielgruppe (Schüler, Studierende, Weiterbildung) und den jeweiligen Anbietern (Hochschulen, Archive, Internetdienste, Fernsehsender, Verlage sowie sonstige private Unternehmen).

Die Recherche erfolgte mit Hilfe von Literaturlieferanten, Fachforen und Onlineverzeichnissen sowie in Sozialen Medien (insb. *twitter*, *digit*, *delicio.us*) zwischen den Jahren 2012 und 2015 (vgl. Auflistung der Recherchequellen im Anhang A.4, S. 268). Eine Aufstellung aller Informationsträger einschließlich Name, Anbieter, Quelle und Gruppenzuordnung gemäß Tab. 4.4 findet sich im Anhang A.5 (S. 269).

Die Informationsträger wurden anhand von vier Kriterien ausgewählt: (1) der Nutzung für Lernvideos, (2) der freien Zugänglichkeit, (3) eine ausreichende Dokumentation und (4) der Einordnung als eine nicht-triviale Lösung.

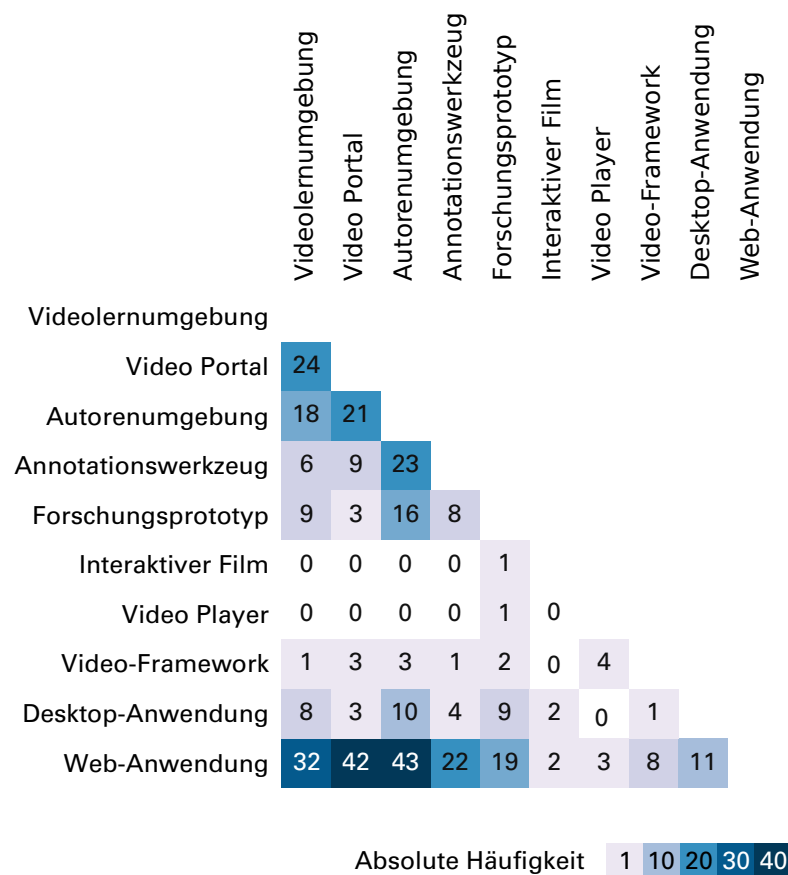
Einige Informationsträger sind nicht allein für die Wiedergabe, Distribution oder Bearbeitung von Lernvideos bestimmt. Die Anforderung bestand darin, dass Anwendungsszenarien im Sinne des Lernens mit Videos existieren. Der Kontext des Videos als Lernmedium stellt somit ein Kriterium zur Auswahl der Informationsträger dar. Wohlgemerkt beziehen sich die Informationsträger nur zu einem relativ kleinen Teil auf formelle Lernangebote. Die weiteren Kriterien betreffen die Zugänglichkeit, ausreichende Dokumentation und Gestaltungshöhe im Sinne nicht-trivialer Lösungen. Alle vier Auswahlkriterien sind in Tab. 4.3 näher beschrieben und hinsichtlich der Kriterienprüfung erläutert.

Die ermittelten Informationsträger lassen sich in acht Gruppen einteilen (Tab. 4.4). Mehrheitlich handelt es sich bei ihnen um Web-Anwendungen. Da sich viele Informationsträger nicht eindeutig einer einzigen Gruppe zuordnen lassen, sind Mehrfachzuordnungen vorgesehen. Beispielsweise handelt es sich bei *DIVER<sup>TM</sup>/WebDIVER<sup>TM</sup>* [Pea und Lindgren 2008; Pea 2006]

<sup>17</sup>Vgl. Internet Archive: <https://archive.org/web/web.php> (abgerufen am 15.02.2015).

um eine audiovisuelle Lernumgebung, die als Online- und Desktop-Anwendung vorhanden ist und aus einem Forschungsprototyp hervorgegangen ist. In der Datengrafik in Abb. 4.1 sind diese Überlappungen quantifiziert.

Durch den beträchtlichen Umfang an identifizierten Lernumgebungen hat sich eine theoretische Sättigung in Bezug auf die Gewinnung neuer Erkenntnisse eingestellt. Die Sättigung resultiert einerseits daraus, dass keine neuartigen Lösungsvarianten der nachfolgend aufgezählten Kategorien gefunden wurden und zu fast jeder Kategorie mindestens drei Beispielanwendungen benannt werden konnten. Der Umfang der Grundgesamtheit bedingte zudem die Möglichkeit einer vollständigen Analyse, ohne eine Stichprobe auswählen zu müssen.



**Abbildung 4.1:** Mehrfachzuordnungen der Gruppen von Informationsträgern



**Tabelle 4.3:** Kriterien zur Auswahl der Informationsträger**Kontext Lernvideo**

*Begründung:* In Anbetracht der Zielstellung dieser Untersuchung bilden videobasierte Lernumgebungen respektive Anwendungen für Lernvideos den Untersuchungsgegenstand und sind ein notwendiges Kriterium der Auswahl von Informationsträgern.

*Kriterienprüfung:* Informationsträger müssen nicht ausschließlich für Lernvideos genutzt werden. Es bedarf jedoch eines Belegs dafür, dass sie zum Zwecke des Lernens im weiteren Sinn nutzbar sind und beispielsweise den Gegenstand informeller Lernprozesse darstellen.

**Freie Zugänglichkeit**

*Begründung:* Ein freier Zugang zum Informationsträger ermöglicht eine umfassende Untersuchung hinsichtlich eigener Untersuchungskriterien und einen Vergleich mit anderen Informationsträgern.

*Kriterienprüfung:* Als frei zugänglich wird ein Informationsträger angesehen, wenn die Benutzung mittels eines gängigen Web-Browser (*Mozilla Firefox, Google Chrome / Chromium, Microsoft Internet Explorer / Edge, Opera, Vivaldi*) kostenfrei im WWW möglich ist bzw. eine kostenfreie Applikation auf einem gängigen Betriebssystem (Windows, Mac OS, Linux-Derivate) installiert werden kann. Das Kriterium gilt auch dann als erfüllt, falls dem Autor für seine Untersuchung der Zugang zu einer Anwendung explizit eingeräumt wird. Bereits abgeschaltete Dienste gelten als zugänglich, sofern ihr ehemaliges Angebot mit Hilfe der *Way Back Machine*<sup>17</sup> nachvollzogen werden kann. Sollte dieses Kriterium nicht erfüllbar sein, bedarf es einer ausführlichen schriftlichen, audiovisuellen oder bildlichen Dokumentation des Informationsträgers.

**Ausreichende Dokumentation**

*Begründung:* Ohne eine ausreichende Beschreibung eines Informationsträgers ist eine Kodierung nicht möglich.

*Kriterienprüfung:* Ein Informationsträger wird als ausreichend dokumentiert angesehen, wenn die Funktionalitäten und das Interaktionsdesign durch Texte und Bilder so ausführlich beschrieben wird, dass ein umfassender Gesamteindruck der Anwendung entsteht. In der Regel ist einer Dokumentation durch einschlägige, wissenschaftliche Publikationen genüge getan. Falls eine Anwendung dem Autor im Untersuchungszeitraum frei zugänglich ist und ihre Funktionalitäten sowie ihr Interaktionsdesign praktisch erfahrbar sind, so wird eine zusätzliche schriftliche bzw. bildliche Dokumentation nicht als notwendig erachtet.

**Nicht-triviale Lösung**

*Begründung:* Da sich die Anzahl an existierenden Videoanwendungen nicht genau beziffern lässt, bedarf es einer Fokussierung auf diejenigen Informationsträger, aus deren Analyse ein Erkenntnisgewinn zu erwarten ist.

*Kriterienprüfung:* Eine Lösung wird als trivial angesehen, wenn a) die Funktionalitäten einer Anwendung nicht über die basalen Steuerungselemente *Play, Pause, Stopp, Zeitleiste, Vollbild, Lautstärkeregelung* hinausgehen oder b) sie sich in Form und der Kombination ihrer Funktionalitäten nicht von einer großen Zahl bereits untersuchter Informationsträger unterscheidet. Als trivial wird beispielsweise eine Gruppe von Videolernumgebungen angesehen, die alle den gleichen Typ eines bestimmten Videoplayers integrieren und darüber hinaus keine Besonderheiten aufweisen.

**Tabelle 4.4:** Gruppen von Informationsträgern

Gruppe	Beschreibung	Anzahl
Videolernumgebung	E-Learning-Anwendung im engeren Sinne, deren Lernressourcen zu einem Teil oder vollständig aus Videos bestehen. Beispiel: <i>udacity</i> <sup>a</sup> .	38
Videoportal	Eine Distributionsplattform für eine große Anzahl an Videos, die durch bestimmte Institutionen oder Einzelanwender beigetragen werden. Im letzteren Fall kann daraus ein <i>social network</i> hervorgehen. Beispiel: <i>YouTube EDU</i> <sup>b</sup> .	55
Autorenumgebung	Eine Anwendung zur Nachbearbeitung bzw. Anreicherung von Videos durch einen oder mehrere Autoren. Beispiel: <i>Hyperfilm</i> [Srl 2007]	54
Video-Framework	Software Framework mit dessen Hilfe sich videobasierte Anwendungen entwickeln lassen. Beispiel: <i>Timesheet.js</i> [Cazenave et al. 2011]	10
Interaktiver Film	Dokumentarischer oder fiktionaler Film, der über eine nicht-lineare Navigationsstruktur verfügt und mit Zusatzmaterialien angereichert ist. Beispiel: <i>Amazonie</i> <sup>c</sup> .	8
Web-Anwendung	Videoanwendung, die im WWW ausgeführt werden kann. Beispiel: <i>MetaVid</i> [Dale et al. 2009]	103
Desktop-Anwendung	Eine Videoanwendung, die nur lokal auf einem Computer ausgeführt werden kann. Beispiel: <i>MediaLoom</i> [Tolva 1998].	16
Forschungsprototyp	Eine prototypische Videoanwendung, die primär zur Untersuchung konkreter Forschungsfragen entwickelt wurde. Beispiel: <i>VideoAnt</i> [ATS 2007]	25
Annotationswerkzeug	Werkzeug zur Analyse, insbesondere zur zeitgenauen Kodierung von Videos im Kontext qualitativer Untersuchungen. Beispiel: <i>Advene</i> [Aubert und Prié 2005]	26
Video Player	Anwendung, die die Wiedergabe von Videos ermöglichen und nur wenige zusätzliche Funktionen aufweisen. Beispiel: <i>Flowplayer</i> <sup>17</sup> .	8

<sup>a</sup>Vgl. <https://www.udacity.com/> (abgerufen am 15.02.2015).<sup>b</sup>Vgl. <http://www.youtube.com/edu> (abgerufen am 15.02.2015).<sup>c</sup>Vgl. <http://amazonie.arte.tv/de/> (abgerufen am 15.02.2015).<sup>17</sup>Vgl. <http://www.flowplayer.org/> (abgerufen am 15.02.2015).

**Bestimmung der Kategorien** Eine notwendige Voraussetzung für die Durchführung einer Inhaltsanalyse ist die Entwicklung eines inhaltsanalytischen Categoriesystems. Die Kategorien sind notwendig, um die Informationsträger systematisch kodieren zu können. Kodierung meint hier, den Merkmalen der Informationsträger (Videolernumgebungen) Kategorien zuzuordnen. Die Qualität der Kodierung hängt somit von der Qualität der Kategorien ab. An ein Categoriesystem stellt Kromrey zwei zentrale Anforderungen: Objektivität und Validität [Kromrey 2011]. Objektivität ist dann gegeben, wenn jeder Sachkundige die Verfahrensweise nachvollziehen kann und die Zuordnung genauso vornehmen würde, sodass „die Resultate der Zuordnung ‘intersubjektive’ Geltung beanspruchen können“ [Kromrey 2011].

Die Forderung nach einem validen Categoriesystem bedeutet in dem Fall, dass unterschiedliche Gestaltungslösungen semantisch richtig zugeordnet werden können (interne Validität) und „die Kategorien korrekte Schlüsse für die Realität zulassen“ [Kromrey 2011].

In Anlehnung an Döring & Bortz wurde das Categoriesystem a priori aus der Literatur deduktiv abgeleitet und während der Durchsicht der Informationsträger ergänzt sowie genauer aufgeschlüsselt, um ein möglichst erschöpfendes Categorieschema zu erhalten [Döring und Bortz 2006, S. 330]. Es wurde darauf geachtet, semantische Überschneidung auszuschließen, um somit eine eindeutige Klassifikation der Informationsträger zu gewährleisten. Die Kategorien genügen zudem dem Anspruch, voneinander unabhängig zu sein, d.h. eine Einordnung eines Merkmals in eine Kategorie bedingt keine weiteren Einordnungen.

Die ermittelten Kategorien lassen sich hinsichtlich ihres funktionalen Einsatzzwecks in sechs Analysedimensionen<sup>19</sup> unterteilen. In der folgenden Aufstellung werden diese Dimensionen kurz beschrieben. Die Zahl in Klammern je Dimension entspricht dabei der Anzahl an untergeordneten Kategorien.

1. *Grundfunktionen (4)*: Grundlegende technische Steuerungs- und Ausgabefunktionen.
2. *Zugang zu zeitbasierter Information (14)*: Umfasst Lösungen, um die durch das Video sukzessive präsentierten bildlichen wie auch akustischen Informationen dem Anwender zugänglich und wiederauffindbar zu machen bzw. den Zugang zu vereinfachen. Eine verbesserte Zugänglichkeit kann durch eine semantische Strukturierung sowie eine Reduktion der kognitiven Last (z.B. aufgrund rascher Bildwechsel) erreicht werden.
3. *Kontribution (8)*: Gestaltungslösungen mit denen Lernende aktiv Inhalte beitragen bzw. vorhandene Informationen bewerten können.
4. *Strukturierung (10)*: Umfasst Lösungen, Lernressourcen nach logischen Kriterien zu ordnen oder zu sortieren sowie zwischen ihnen Beziehungen herzustellen.
5. *Selbstorganisation (7)*: Gestaltungslösungen, die den Lernenden bei der Planung und Bewältigung von Lernaktivitäten sowie bei der Wissensorganisation unterstützen.
6. *Layouts (7)*: Aspekte der räumlichen Anordnung von Objekten innerhalb der zur Verfügung stehenden Fläche.

In Tab. 4.5 sind die insgesamt 50 Kategorien dargestellt und den sechs Analysedimensionen zugeordnet. Weitere Ausführungen zu den einzelnen Kategorien finden sich in den zurückliegenden Kapiteln 2 und 3.

<sup>19</sup>In der Literatur bezeichnet man diese auch als Oberkategorien.

**Tabelle 4.5:** Kategorien je Analysedimension

<b>Grundfunktionen</b>	
basic controls	Verfügbarkeit grundlegender Steuerelemente: Play/Pause, Zeitleiste, Lautstärkeregler, Video-Vollbild.
appropriate delivery	Bestmögliche Videoqualität und -größe in Anbetracht der Übertragungskapazitäten des Netzwerks und Wiedergabeoptionen im Browser.
loading indicator	Gibt Rückmeldung, falls sich die Videowiedergabe z.B. aufgrund Übertragungsproblemen verzögert.
shortcut commands	Steuerung der Anwendung einschließlich der Videowiedergabe per Tastatur, Touch-Gesten oder ähnlichen Kurzbefehlen.
<b>Zugang zu zeitbasierter Information</b>	
search	Suchfunktion innerhalb eines Videos oder innerhalb einer Sammlung von Videos.
table of content	Eine in Bezug auf die Wiedergabezeit chronologische Auflistung von Zwischenüberschriften ermöglicht einen schnellen Zugriff auf die korrespondierenden Inhalte des Videos.
temporal tags	Zeitgenaue semantische Annotation von Schlüsselwörtern in einem Video oder innerhalb einer Sammlung von Videos.
temporal bookmark	Referenz auf einen Zeitpunkt innerhalb der Wiedergabezeit eines Videos.
playback speed	Möglichkeit die Abspielgeschwindigkeit gemäss den persönlichen Erfordernissen anzupassen.
playback direction	Möglichkeit die Richtung der zeitabhängigen Informationsdarbietung zu ändern.
zoom	Vergrößerungsfunktion für visuelle Videoinhalte während der Wiedergabe.
visual summary	Bildliche Zusammenfassung der Schlüsselszenen eines Videos.
annotated timeline	Interaktive grafische Abbildung des zeitlichen Layouts einschließlich der darin vorkommenden Ereignisse und Annotationen. Die abgebildeten zeitgenauen Ereignisse fungieren als Navigationselement.

(Fortsetzung auf der nächsten Seite)

(Tabelle 4.5 – Fortsetzung von der vorherigen Seite)

transcript	Repräsentation des gesprochenen Wortes (ggf. auch Bildbeschreibungen) in Textform, wobei einzelne Passagen mit der korrespondierenden Abspielposition im Video verknüpft sind.
closed captions	Optionale Texteinblendung während der Videowiedergabe, in der gegenwärtig gesprochene Worte und Dialoge oder eine Beschreibung der visuellen Informationen dargestellt sind.
skip back	Einfache Möglichkeit die aktuelle Abspielposition um wenige Sekunden zurückzusetzen, um die letzten Ausführungen (z.B. Bewegungsabläufe, Worte) noch einmal rezipieren zu können.
journal navigation	Berücksichtigung der zeitgenauen Adressierung der Abspielposition beim Vor- und Zurückgehen im Browser.
loop	Möglichkeit zur erneuten Wiedergabe des gesamten Videos oder eines Teils davon.
<b>Kontribution</b>	
add video	Upload, Referenzierung oder Einbettung videografischer Ressourcen.
annotations	Allgemeine Möglichkeit den Inhalt eines Videos durch zeitabhängige und zeitbezogene Inhalte anzureichern.
comments	Zeitgenaue Kommentare, die sich zu verankerten Diskussion entwickeln können.
inline drawing	In einer Schicht oberhalb des Videobildes lassen sich Freihandzeichnungen anfertigen, um zum gegenwärtigen Zeitpunkt der Wiedergabe Objekte zu kennzeichnen oder hervorzuheben.
polls	Die Wiedergabe wird für eine Abstimmung zu einer bestimmten Fragestellung oder einem Thema unterbrochen. Die kumulierten Ergebnisse der Abstimmung werden anschließend dargestellt.
video manipulation	Anwender können Eigenschaften manipulieren, welche die Darstellung des Videos hinsichtlich der Abmessungen (z.B. Seitenverhältnis, Höhe, Breite, Form des Video Frames), Bildeigenschaften (z.B. Farbe, Kontrast) oder sonstiger Bildmanipulationen und -effekte (z.B. „green screen“) betreffen.

(Fortsetzung auf der nächsten Seite)

(Tabelle 4.5 – Fortsetzung von der vorherigen Seite)

direct authoring	Möglichkeit zur unmittelbaren Bearbeitung oder Anreicherung des Videos, ohne dass ein Medienbruch durch die Nutzung einer gesonderten Autorenumgebung entsteht.
multi-timeline editing	Zur Bearbeitung bzw. Annotation des Videos stehen mehrere parallele Spuren (auch Tracks genannt) zur Verfügung, auf denen sich Objekte jeweils zeitlich positionieren lassen.
remix	Arbeitsergebnisse von Dritten (z.B. Annotationen, Überprüfungsfragen oder Arrangements von Videos) bilden die Grundlage einer eigenständigen Bearbeitung des Videos.

**Strukturierung**


---

video manager	Übersicht über alle Videos hinsichtlich eines bestimmten Kriteriums, sodass einzelne Ressourcen bei Bedarf aufrufbar sind.
sequential media	Verkettung von kontinuierlichen und diskreten Medien, die nacheinander (sequenziell) aufgerufen und wiedergegeben werden können.
related videos	Angebot zusätzlicher Lernvideos während oder nach der Wiedergabe eines Videos.
hyperlinks	Referenz auf eine Ressource, die durch einen räumlich und zeitlich positionierten Linkanker im Videobild aufgerufen werden kann.
branching videos	Möglichkeit während der Wiedergabe zwischen alternativen Lernpfaden bzw. Lernvideos wählen zu können.
detail on demand	Auf Anfrage werden Detailinformationen in Form von Videosequenzen angeboten, wobei sich nach deren Wiedergabe das Ausgangsvideo fortsetzt.
media fragments	Möglichkeit zur Wiederverwendung oder Zitation von zeitlich oder räumlich begrenzten Fragmenten eines Videos.
classified marks	Qualitative Unterscheidung einer größeren Anzahl zeitgenauer Ereignisse in einem Video.
user ratings	Möglichkeit quantitative Bewertungen mehrerer Anwender zu erheben und akkumuliert darzustellen.
break	Eingebaute Pause von begrenzter Dauer, in der der Anwender die zurückliegende Szene/Inhaltseinheit wirken lassen kann, bevor die Wiedergabe fortsetzt.

(Fortsetzung auf der nächsten Seite)

(Tabelle 4.5 – Fortsetzung von der vorherigen Seite)

**Selbstorganisation**


---

viewing history	Übersicht der zuletzt bzw. in der letzten Sitzung betrachteten Videos.
playlist	Eine vom Anwender zusammengestellte Vorauswahl an Videos zum Zwecke einer späteren Betrachtung oder Weiterverwendung.
follow revisions	Nachvollziehbare Aufzeichnungen von Änderungen, die an einem Hypervideo-Dokument durch verschiedene Anwender vorgenommen wurden.
user notes	Anwender können Notizen zeitgenau mit Inhalten eines Video verknüpfen.
assessment	In Abhängigkeit der Wiedergabezeit sind Überprüfungsfragen in ein Video integriert.
user traces	Darstellung von Nutzeraktivitäten und -verhalten unter Berücksichtigung der Wiedergabezeit.

**Layout**


---

full screen	Integration zeitabhängiger Medien und essentieller Steuerungselemente in einer Vollbildansicht.
simultaneous media	Die zeitgleiche Darstellungen verschiedener Medienströme, sodass diese sich nicht gegenseitig überdecken.
synchronized map	Synchrone und parallele Kartendarstellung von geografischen Orten, Strecken oder Gebieten in Abhängigkeit der Wiedergabezeit.
overlays	Vollständige oder teilweise Überlagerung des Videobildes in Abhängigkeit der Wiedergabezeit zur Darstellung zusätzlicher Inhalte. Wird zeitabhängig oder durch eine Benutzerinteraktion ausgelöst.
visual highlighting	Zeitlich und räumlich begrenzte, visuelle Hervorhebungen einzelner Elemente in einem Videobild.
object tracking	Automatische Platzierung von Annotationen gemäß der Position von bewegten Objekten im Video.
multi-angle video	Der Betrachter kann zwischen mehreren Kameraperspektiven wählen oder diese in Form von separaten Videos gleichzeitig betrachten.

---



**Analyse der Informationsträger** Die Analyse der 121 Informationsträger fand im Januar und Februar 2015 statt, wobei auf einige Aufzeichnungen vom September 2012 und Januar 2014 aufgebaut werden konnte. Die Analyse wurde zunächst vom Autor allein durchgeführt und in englischer Sprache verfasst. Im Anschluss bewertete eine zweite, unabhängige Person eine Teilmenge der Informationsträger.

Im ersten Schritt wurden die im Anhang A.6 (S. 287) verzeichneten Metadaten in einer Datenbank dokumentiert (siehe auch Abb. 4.2). Bei der Datenbank handelt es sich um eine vom Autor für diesen Zweck entwickelte Web-Anwendung, die im Laufe dieses Kapitels noch vorgestellt wird (vgl. Teilabschnitt 4.3.3, S. 145).

Name	VideoAnt
Description	<p>VideoAnt is an academic video annotation tool that enables users to work with referenced video sources (e.g. mp4 or videos from YouTube). These wrapped up videos can be annotated with a subject heading and a text comment to a certain position in time. The screen is divided by the timeline on top and the video next to the comment list below. The video can be controlled by the standard YouTube control elements. To add a comment a button below the video frame can be pressed. Each comment is visualized on the timeline above the video and comment list. On hovering the the marks on the timeline the title of the comment appears. Clicking on them moves the playback to the temporal position and highlights the the comment in the list. Other user can comment on these annotations (nested comments). By selecting a comment box the playback proceeds at the corresponding position. It is possible to edit or delete a previously annotated comment.</p> <p>Export functions for RSS, JSON, XML, and text are provided aside common sharing and embedding facilities. All date is stored on the server. Access is provided through third party accounts at google, facebook, twitter or shibboleth.</p>
Provider	Academic Technology Services, College of Education & Human Development, University of Minnesota
URL	<a href="https://ant2.cehd.umn.edu/">https://ant2.cehd.umn.edu/</a>
Tags	Video Authoring Environment,Online Environment,Research Prototype,Annotation Tool
Patterns	Add Video,Comments,Annotations,Annotated Timeline
Availability	Online available.
Accessed personally? <input checked="" type="checkbox"/>	
<input type="button" value="update"/> <a href="#">cancel</a> <a href="#">delete</a>	

**Abbildung 4.2:** Exemplarische Eingabemaske eines kodierten Informationsträgers

Die tiefgründige Analyse der Informationsträger bildete den zweiten Schritt. Zunächst wurde ein *drill-down*-Verfahren (vgl. van Welie und Klaasse [2004]) angewandt, um die Informationsträger abschnittsweise oder seitenweise (bei Webseiten) zu betrachten, sofern diese einen Bezug zu Videos hatten. Für jeden relevanten Abschnitt / jede relevante Seite einer Anwendung wurde geprüft, inwiefern die darin vorkommenden Interaktions- und Strukturmerkmale den in Tab. 4.5 definierten Kategorien semantisch entsprechen. Uniforme, sich wiederholende Seitenstrukturen (z.B. innerhalb von Portalen) wurden nur stichprobenweise ausgewählt. Bislang nicht betrachtete Interaktionsformen wurden als Kandidaten für zusätzliche Analysekategorien erfasst und in die Analyse einbezogen. Als hilfreich bei der Suche nach Merkmalen erwiesen sich dabei Produktmerkmalsbeschreibungen, Blog-Beiträge der Unternehmen sowie wissenschaftliche Publikationen der betrachteten Anwendungen oder Prototypen. Nicht mehr verfügbare, teils historische Anwendungen wurden, soweit möglich, mit Hilfe der *Way Back Machine* sowie mit Hilfe von im WWW verfügbaren *screencasts* rekonstruiert.

Die Merkmalsausprägung einer jeden identifizierten Analysekategorie wurde in Wort und Bild dokumentiert. Mit Hilfe von einem oder mehreren *screenshots* ließ sich das Design der Interaktions- und Strukturelemente weitestgehend erfassen. Teilweise enthalten die *screenshots* Merkmale verschiedener Kategorien. In nur wenigen Fällen war es notwendig ein Bildschirmvideo (*screencast*) aufzuzeichnen, um komplexe dynamische Visualisierungen (z.B. Animationen) infolge von Benutzerinteraktionen festzuhalten. Insgesamt wurden 594 *screenshots* und 4 *screencasts* erstellt und den Informationsträgern sowie Kategorien in der Datenbank zugeordnet.

In derselben Datenbank erfolgte auch die Beschreibung der Informationsträger einschließlich der Merkmalsausprägungen je identifizierter Kategorie. In Abb. 4.2 ist die Eingabemaske dargestellt. Neben dem Nutzungskontext des Informationsträgers und Angaben zur Zielgruppe wurden insbesondere die relevanten Teile der Informationsstruktur erfasst. Teilweise wurde auch das Layout von Ansichten bzw. Unterseiten, die Instanzen einer Analysekategorie enthielten, festgehalten. Die Funktionsweise relevanter Interaktionselemente bildete jedoch den Schwerpunkt der Beschreibungen. Unabhängige Teilanwendungen einzelner Informationsträger wurden dabei in gesonderten Absätzen behandelt.

Zur Sicherstellung der Intersubjektivität des Categoriesystems führte eine zweite Person eine unabhängige Kategorisierung durch. Bei diesem Bewerter handelte es sich um einen erfahrenen Mediendesigner, der seit mehreren Jahren multimediale Lernumgebungen entwirft. Um den Zeitaufwand für die Analyse des Bewerter in einem vertretbaren Maß zu halten, wurde eine randomisierte Teilmenge von 25 Informationsträgern (20,66 % der Grundgesamtheit) ermittelt. Die Analyse der Interrater-Reliabilität hinsichtlich der Anzahl übereinstimmender und nicht übereinstimmender Kategorien ergab beachtliche bis fast vollkommene Übereinstimmungen, sowohl für die betrachteten Informationsträger, als auch für die Kategorien. Die Werte für Cohens Kappa ( $\kappa$ ) lagen zwischen 0,72 und 1,00 [Cohen 1960; Landis und Koch 1977] (vgl. Anhang A.7, S. 288).

**Ergebnisse und Diskussion** Im Ergebnis dieser Analyse steht die in Tab. 4.7 dargestellte, nach ihrer Auftrittshäufigkeit sortierte Liste der Analysekategorien. Alle Informationsträger ließen sich durch die Kategorien beschreiben. Allerdings konnten zwei Kategorien (*multi-angle video* und *playback direction*) in keiner der untersuchten Lernumgebungen identifiziert werden. Es stellt sich hier die Frage, inwieweit ausreichend Informationsträger in die Analyse einbezogen wurden, zumal der Großteil von angloamerikanischen und europäischen Anbietern stammt. Dies hat mehrere Ursachen. Zum einen sind Autoren aus Afrika, Südamerika und Asien in den angesehenen Journalen und Tagungsbänden (*Proceedings*) unterrepräsentiert (vgl. z.B. Thomson Reuters [2011]), zum anderen geben Suchmaschinen wie Google personalisierte, d.h. auch geozentrische Ergebnisse aus. Des Weiteren konnte der Autor Quellen, die nicht in englischer, deutscher oder spanischer Sprache verfasst waren, nur eingeschränkt (z.B. mittels maschineller Übersetzung im Web-Browser) auswerten. Dieser geografische Bias wurde versucht durch die Auswertung von *Social Bookmarking* sowie *Microblogging Diensten* zu kompensieren, die von Anwendern fast aller Erdteile gleichermaßen genutzt werden<sup>20</sup>. Dennoch kann kein Anspruch auf Vollständigkeit erhoben werden. Die dynamische Entwicklung von *Start-Ups* in dem Bereich der Bildungstechnologien sowie die technologischen und wissenschaftlichen Fortschritte bedingen vielfache Weiterentwicklungen von Lernumgebungen sowie das Emporkommen und Verschwinden von Angeboten.

Auch in Bezug auf das verwendete Categoriesystem ist eine Vollständigkeit nur schwer zu belegen, jedoch gibt das Urteil des zweiten Bewerbers Grund zu der Annahme, ein umfassendes Set an Kategorien genutzt zu haben.

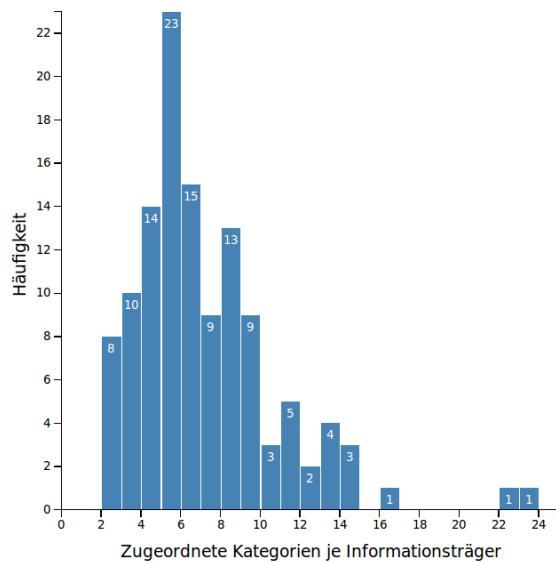
Für die 121 untersuchten Informationsträger  $I = i_1, i_2, \dots, i_{121}$  und den zugeordneten Kategorien  $K = k_1, k_2, \dots, k_{50}$  wird folgende Metrik entwickelt, um das Ergebnis mit Hilfe von Kennwerten besser beschreiben zu können. Sei  $K_i = k_{i,1}, k_{i,2}, \dots, k_{i,n}$  die Menge der Kategorien eines Informationsträgers  $i \in I$ , wobei  $K_i \in K, n \leq 50$ . Die Menge der Informationsträger mit der Kategorie  $k \in K$  sei  $I_k = i_{k,1}, i_{k,2}, \dots, i_{k,m}$ , wobei  $I_k \in I, m \leq 121$ .

Als Kennwert wird die Funktionalitätsrate  $u_i$  als das Verhältnis der Anzahl der einem Informationsträger zugeordneten Kategorien  $K_i$  zur Anzahl der bestehenden Kategorien  $K$  definiert, wobei  $u_i \in [0, 1]$ :

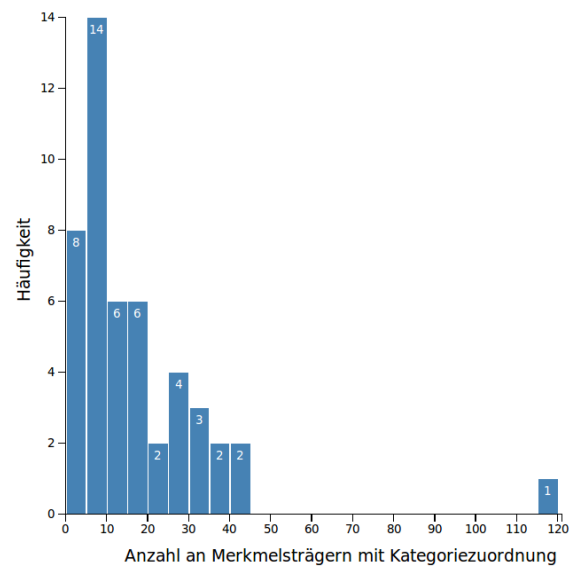
$$u_i = \frac{|K_i|}{|K|} \quad (4.1)$$

Das Histogramm in Abb. 4.3 zeigt die Häufigkeitsverteilung der je Informationsträger identifizierten Kategorien. Den untersuchten Videosystemen wurden zwischen 2 und 23 Kategorien zugewiesen. Der Modus lag hier bei fünf Kategorien. Für die untersuchten Videolernumgebungen liegt die Funktionalitätsrate zwischen 0,04 und 0,44. Dies zeigt, dass keine universelle Plattform existiert, in der sämtliche Features realisiert sind. Nur 17 der 121 Plattformen wiesen mehr als zehn Kategorien auf. Spitzenreiter im Feld ist *Theresienstadt explained* sowie *YouTube* mit Merkmalsausprägungen in 22 Kategorien, gefolgt von *Matterhorn* (16) sowie *Kaltura* (14). Diese quantitativen Angaben erlauben jedoch keine qualitativen Aussagen hinsichtlich der jeweiligen Lernumgebungen, da die einzelnen Merkmale sehr unterschiedlich ausgestaltet sind und die *Usability* und *User Experience* keine Berücksichtigung fanden.

<sup>20</sup>Vgl. u.a. <http://tweetping.net/> (abgerufen 23.05.2016).



**Abbildung 4.3:** Verteilung der Anzahl der pro Informationsträger identifizierten Kategorien



**Abbildung 4.4:** Verteilung der Anzahl an Informationsträgern je Kategorie

Der Verbreitungsgrad oder die relative Häufigkeit  $v_k$  beschreibt den Anteil der Informationsträger, denen eine Kategorie  $k \in K$  zugewiesen werden konnte, wobei  $v_k \in [0, 1]$ .

$$v_k = \frac{|I_k|}{|I|} \quad (4.2)$$

In der nachfolgenden Tab. 4.7 (S. 127) ist der Verbreitungsgrad der Kategorien als relative Auftrittshäufigkeit in Prozent dargestellt. Die in Abb. 4.4 dargestellte Häufigkeitsverteilung der ermittelten Informationsträger je Kategorie erstreckt sich von 1 bis 121. Die Kategorie *Basic Controls*, welche allen 121 Informationsträgern zugeordnet werden konnte, bildet die obere Grenze. Im unteren Bereich ist zwischen 1 und 10 Zuordnungen die größte Häufung zu verzeichnen, d.h. die meisten Informationsträger verfügen über weniger als 10 Kategorien und somit über vergleichsweise wenige Funktionen. 26 der 50 Kategorien konnten demnach in weniger als 10 Anwendungen identifiziert werden. Fast die Hälfte der Kategorien ließ sich somit nur einer kleinen Anzahl von Anwendungen zuschreiben. 24 Kategorien finden sich in weniger als 10 % der betrachteten Informationsträger. Die Gründe für diese geringe Verbreitung sind vielfältig. Zum einen sind die Kategorien mit sehr geringer Auftrittshäufigkeit technisch nicht so einfach zu realisieren (z.B. *zoom*). Zum anderen handelte es sich vielfach um schlanke Anwendungen, die für spezifische Einsatzszenarien und Zielgruppen mit dem notwendigen Set an Funktionalitäten ausgestattet wurde. Eine weitere Ursache resultiert aus den jeweiligen technischen Möglichkeiten zum Zeitpunkt der Entwicklung.

Um eine Kategorie als ein Pattern bezeichnen zu können, ist zu belegen, dass es sich um eine wiederkehrende Lösung für ein verbreitetes Problem handelt. Die Kategorien bilden somit lediglich Hypothesen, die durch die Beispielinstanzen zu belegen sind. Kann nur eine Anwendung gefunden werden, gilt die Gestaltungslösung als Zufall. Gibt es jedoch zwei Anwendungen der Lösung von unterschiedlichen Entwicklern, spricht man von einem „Pattern Candidat“ [Kohls und Panke 2010; Vlissides 1996]. In der Pattern Community werden instantiierte Lösungen je-

doch erst dann als Pattern angesehen, wenn mindestens drei Anwendungen unterschiedlicher Entwickler belegt werden können („Rule of Three“) [Kohls und Panke 2010; Kohls und Uttecht 2009; Seffah 2010]. Eine Selektion ist außerdem empfehlenswert, um die Nutzer der Pattern Language nicht durch eine übermäßig hohe Anzahl an Patterns zu überfordern [Cutumisu et al. 2006].

45 Kategorien ließen sich in mehr als drei Informationsträgern identifizieren, während 5 Kategorien nur in einem, zwei oder keinem Informationsträger zu finden waren.

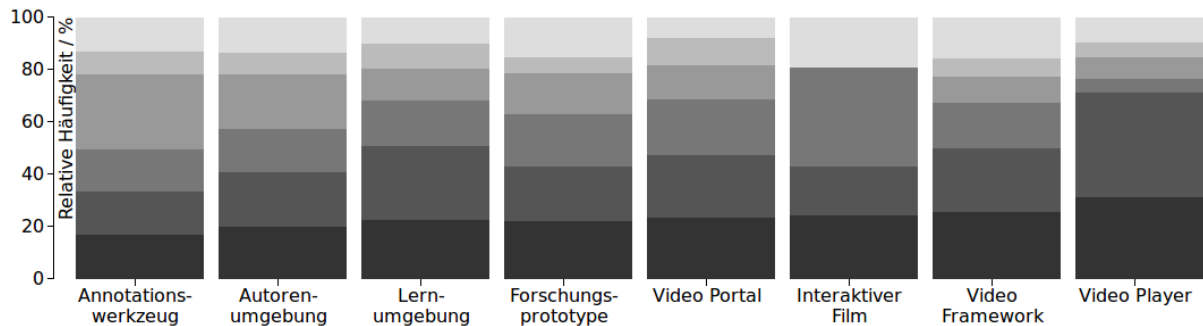
**Tabelle 4.6:** Vergleich der Analysedimensionen mit Angabe der Anzahl an zugehörigen Kategorien. Absolute und relative Häufigkeit (%), arithmetisches Mittel ( $M$ ) und Standardabweichung ( $SD$ ) bezogen auf die Kategorien und identifizierten Merkmalsausprägungen je Dimension.

	Kategorien				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	$M$	$SD$
Grundfunktionen	4	192	23.30	48.00	43.49
Kontribution	9	123	14.93	13.67	12.88
Zugang zu zeitbasierter Information	13	187	22.69	14.38	11.19
Strukturierung	10	160	19.42	16.00	10.23
Layout	6	89	10.80	14.83	12.25
Selbstorganisation	6	73	8.86	12.17	6.52
Gesamt	48	824	100	17.17	18.87

Ein Blick auf die Ergebnisse in den einzelnen Analysedimensionen zeigt eine unterschiedliche Verteilung hinsichtlich der Häufigkeit und dem arithmetischem Mittel der kodierten Kategorien (vgl. Tab. 4.6). Betrachtet man die Häufigkeiten der Kodierungen je Dimension, dominieren die Grundfunktionen sowie strukturierende Elemente auf Makro- und Mikrolevel über die Hälfte des Feldes. Etwas mehr als ein Drittel der 819 Merkmalsausprägungen steht mit dem Layout, der Kontribution und Selbstorganisation in Zusammenhang. Dieses Bild bestätigt sich etwas weniger deutlich in Bezug auf das arithmetische Mittel der pro Kategorie identifizierten Merkmalsträger. Das vergleichsweise hohe arithmetische Mittel (48,00) der Grundfunktionen resultiert aus der hohen Verbreitung der *basic controls*.

Die bislang dargestellten Ergebnisse ermöglichen noch keine Aussagen hinsichtlich der unterschiedlichen Gruppen von Informationsträgern (vgl. Tab. 4.1). Wie bereits bemerkt, sind die Gruppen nicht trennscharf im Sinne einer Klassifikation. Aus diesem Grund sind Korrelationen zwischen Gruppen und den Kategorien oder Dimensionen nur dann zulässig, wenn man die Gruppen als *Fuzzy-Mengen* festlegen würde. Dennoch lässt sich die Häufigkeitsverteilung der Kategorien in den Analysedimensionen je Gruppe von Informationsträgern gegenüberstellen (vgl. Abb. 4.5). Eine auffällige Varianz hat die Dimension der Nutzerkontributionen, welche in den Lernumgebungen und Videoportalen weniger häufig vertreten sind, als in Annotationswerkzeugen. Funktionen zur Unterstützung des selbstgesteuerten bzw. selbstorganisierten Ler-

nens finden sich in allen Dimensionen von Lernumgebungen nur zu einem geringen Anteil. Die für das Web 2.0. typischen Funktionen sind demnach nicht sehr verbreitet und wurden nur selten für kontinuierliche Medien adaptiert. Ein Detailbetrachtung je Gruppe ist im Anhang A.8 (S. 291) hinterlegt. Gleiches gilt für weiterführende Analysen des hier nur vor dem Hintergrund der Identifikation von Patterns betrachteten Datensatzes (vgl. Anhang A.8, S. 294).


























**Abbildung 4.5:** Häufigkeitsverteilung der Kategorien in den Analysedimensionen je Gruppe von Informationsträgern: ■ Layout, ■ Selbstorganisation, ■ Kontribution, ■ Strukturierung, ■ Zugang zu zeitbasierter Information, ■ Grundfunktionen

**Tabelle 4.7:** 45 Kategorien von UI-Komponenten von Videolernumgebungen je Dimension (Dim.), sortiert nach der Auftrittshäufigkeit in den untersuchten Informationsträgern. Die hellgrauen Balken kennzeichnen Kategorien auf dem Level der Mikrointeraktivität und die dunkelgrauen Balken jene der Makrointeraktivität.

Dim.	Merkmalskategorien	Absolute und relative Häufigkeit
1	<i>Basic Controls</i>	121 // 100,00 %
3	<i>Add Video</i>	42 34,71 %
1	<i>Loading Indicator</i>	41 33,88 %
2	<i>Annotated Timeline</i>	35 28,93 %
2	<i>Search</i>	35 28,93 %
4	<i>Video Manager</i>	33 27,27 %
6	<i>Simultaneous Media</i>	33 27,27 %
6	<i>Overlays</i>	31 25,62 %
3	<i>Annotations</i>	29 23,97 %
2	<i>Table of Content</i>	28 23,14 %
4	<i>Sequential Media</i>	27 22,31 %
4	<i>Hyperlinks</i>	27 22,31 %
4	<i>Related Videos</i>	24 19,83 %
5	<i>Playlist</i>	21 17,36 %
5	<i>User Traces</i>	18 14,88 %
2	<i>Closed Captions</i>	18 14,88 %
2	<i>Transcript</i>	17 14,05 %
1	<i>Appropriate Delivery</i>	17 14,05 %
5	<i>Assessment</i>	16 13,22 %
3	<i>Comments</i>	15 12,40 %
2	<i>Playback Speed</i>	14 11,57 %
4	<i>Detail on Demand</i>	13 10,74 %
3	<i>Direct Authoring</i>	13 10,74 %
1	<i>Shortcut Commands</i>	13 10,74 %
4	<i>Classified Marks</i>	12 9,92 %

(Fortsetzung auf der nächsten Seite)

(Tabelle 4.7 – Fortsetzung von der vorherigen Seite)

Dim.	Merkmalskategorien	Absolute und relative Häufigkeit	
3	<i>Multi-Timeline Editing</i>	11 	9,09 %
2	<i>Visual Summary</i>	10 	8,26 %
4	<i>User Ratings</i>	9 	7,44 %
2	<i>Temporal Bookmarks</i>	9 	7,44 %
6	<i>Object Tracking</i>	8 	6,61 %
5	<i>Viewing History</i>	8 	6,61 %
4	<i>Media Fragments</i>	7 	5,79 %
5	<i>User Notes</i>	7 	5,79 %
6	<i>Visual Highlighting</i>	7 	5,79 %
6	<i>Full Screen</i>	7 	5,79 %
2	<i>Temporal Tags</i>	6 	4,96 %
4	<i>Branching Videos</i>	6 	4,96 %
3	<i>Remix</i>	5 	4,13 %
3	<i>Video Manipulation</i>	5 	4,13 %
2	<i>Skip Back</i>	5 	4,13 %
2	<i>Journalized Navigation</i>	4 	3,31 %
2	<i>Loop</i>	3 	2,48 %
6	<i>Synchronized Map</i>	3 	2,48 %
2	<i>Zoom</i>	3 	2,48 %
5	<i>Follow Revisions</i>	3 	2,48 %
3	<i>Polls</i>	2 	1,65 %
4	<i>Break</i>	2 	1,65 %
3	<i>Inline Drawing</i>	1 	0,83 %
1	<i>Multi-Angle Video</i>	0	0,00 %
2	<i>Playback Direction</i>	0	0,00 %

### 4.2.3 Analyse und Integration ähnlicher Muster bzw. Mustersprachen

Während eine Mustersprache die Beziehungen der enthaltenen Muster zum Ausdruck bringt, stehen auch Mustersprachen bestimmter Anwendungsgebiete miteinander in Beziehung. Dies ist insbesondere dann der Fall, wenn sich die Anwendungsgebiete überschneiden. Auf die Möglichkeit der Überlappung von Mustern unterschiedlicher Sprachen wies auch Alexander [1979, S. 329] hin. In vielen Mustersprachen finden sich daher Verweise auf Muster anderer Autoren (z.B. Buschmann und Henney [2008]; Crumlish und Malone [2009]; Pratt und Winters [2007]; Schümmer und Lukosch [2007]; Yahoo! [2012]). Im zweiten Schritt des Pattern Minings geht es deshalb darum, Beziehungen zu anderen Mustersprachen zu identifizieren, von ihnen gegebenenfalls zu abstrahieren oder sie zu integrieren. Gegenstand der Analyse sind Musterkollektionen ähnlicher Anwendungsgebiete. Nachfolgende Tabelle 4.8 enthält eine Aufstellung derjenigen Mustersprachen, die aufgrund ihrer Originalität und Aktualität als relevant erachtet wurden. Die Analyse zielt dabei auf zwei Aspekte: einen strukturellen Vergleich der Mustersprachen und die Integration oder Adaption von Mustern.

Da die Strukturanalyse bereits in Kapitel 4.1.2 (S. 103) erfolgt ist, soll an dieser Stelle nur auf das Resultat der konstruierten Mustersprache in Abschnitt 4.3 (S. 140) verwiesen werden.



**Tabelle 4.8:** Pattern Languages, aus denen Muster adaptiert werden konnten

Pattern Language	Themenbereich	Quelle(n)
Designing Interfaces: Patterns for effective Interaction Design	Interaction Design	Tidwell [2005, 2011]
User-Centered Interaction Design Patterns for Interactive Digital Television Applications	interaktives Fernsehen	Kunert [2009]
Patterns for Time-based Hypermedia Artifacts	Hypermedia Design	Lopes und Carriico [2007]
Patterns for Computer-Mediated Interaction	CSCW	Schümmer und Lukosch [2007]
Patterns in Interaction Design	Interaction Design	van Welie und Veer [2003]
Multimedia-Reuse Pattern Language	Multimedia	Cybulski und Linden [1998]
Yahoo Design Pattern Library	Web Design	Yahoo! [2012]
Quince UX Paterns	Interaction Design	Infragistics [2013]

Integration meint in diesem Zusammenhang die Einbeziehung von Mustern anderer Mustersprachen. Zwischen einem hier identifizierten Pattern und einem Muster einer anderen Sprache sind sechs Beziehungen zu differenzieren:

- **Verfeinerung:** Ein Muster stellt eine weiterentwickelte Lösung eines anderen Musters dar, indem es ein Teilproblem adressiert [Buschmann und Henney 2008]. Das Muster VIDEO MANAGER auf S. 333 ist beispielsweise eine Weiterentwicklung des PICTURE MANAGER von Tidwell [2005, S. 40], aber dennoch durch die Inhaltsanalyse empirisch belegt. Siehe auch *Specialization* [Retalis et al. 2006; van Welie und Veer 2003].
- **Abstraktion:** Im Gegensatz zur *Verfeinerung*, ist es auch möglich eine Abstraktion von verwandten Mustern in die Mustersprache zu integrieren. Das Muster VIDEOCLIPQUEST [Kortenkamp und Blessing 2011] beschreibt eine spezifische Form des videobasierten Assessments und wurde deshalb im Muster ASSESSMENT verallgemeinert und als ein Beispiel integriert. Siehe auch *Aggregation* [Retalis et al. 2006; van Welie und Veer 2003].
- **Komplementäre Muster:** Die Gemeinsamkeit komplementärer Muster liegt im Kern eines Problems. Die Unterschiede zeichnen sich durch den Lösungsweg und demzufolge auch durch die damit einhergehenden Wechselwirkungen im Kontext aus. Beispielsweise sind die Problembeschreibungen ANNOTATED TIMELINE (vgl. S. 321) und ANNOTATED SCROLLBAR [Tidwell 2005] insofern analog, dass in Jenifer Tidwells Muster Navigationswerkzeuge für das spatiale Layout und in Bezug auf Videos für das zeitliche Layout mit Annotationen angereichert werden. [Retalis et al. 2006]. Siehe auch *Association* [van Welie und Veer 2003].

- **Identität:** Theoretisch ist es möglich, dass identische Muster in verschiedenen Domänen ihren Platz in der jeweiligen Mustersprache besitzen [Martin et al. 2001, S. 48]. In solch einem Fall muss man abwägen, ob die Identität allein durch einen Verweis oder zum besseren Verständnis vollständig zitiert werden soll. Die bestehenden Patterns *LOADING INDICATION* von Kunert [2009] sowie *LOADING INDICATOR* und *PROGRESS BAR* von Tidwell [Tidwell 2005, 2011] propagieren sehr ähnliche Lösungen für ein nahezu identisches Problem, was auch im hier identifizierten und namentlich übereinstimmenden Muster *LOADING INDICATOR* zum Ausdruck kommt. Der Unterschied zwischen den vier Mustern besteht jedoch im Spannungsfeld des Problems (innerhalb der Pattern Language) sowie in den Auswirkungen der umgesetzten Lösung.
- **Verbund:** Zwei Muster bilden eine größere Struktur, die ein übergeordnetes, komplexeres Problem adressiert [Buschmann und Henney 2008]. Siehe auch *Association* [van Welie und Veer 2003].

Gegenstand der Analyse der acht Pattern Languages waren insgesamt 547 Patterns. Für nur 65 dieser Patterns konnte mindestens eine Art von Beziehung zu den hier gegenständlichen Design Patterns für Videolernumgebungen nachgewiesen werden. Insgesamt wurden dabei 80 Beziehungen identifiziert. In 48 Fällen handelte es sich bei den Video Patterns um eine Verfeinerung der untersuchten Patterns, welche keinen Bezug zu zeitkontinuierlichen Medien aufwiesen. Beispielsweise erstreckte sich die Navigation auf die zweidimensionale Fläche des Bildschirms, nicht jedoch auf die Zeit als dritte Dimension. Des Weiteren wurde auf die Manipulation von Text und statischen Bildern, nicht jedoch auf die Spezifika der Video- oder Audiotbearbeitung eingegangen. In elf Fällen wurden Muster gefunden, die konkretere Lösungen vermitteln, als sie in den Video Patterns beschrieben sind (Abstraktion). Für sechs Video Patterns wurden in sieben Fällen fast identische Lösungen in einem anderen Kontext identifiziert, auf die in den Pattern Beschreibungen später unbedingt verwiesen werden sollte. Lediglich zwei Muster stellen komplementäre Lösungen dar. Zwölf übergreifende Verbunde zwischen den untersuchten und den Video-Patterns wurden ausschließlich bei Lopes und Carriico [2007] gefunden. Eine vollständige Darstellung der Analyseergebnisse ist im Anhang A.9 (S. 297) hinterlegt.

Zum einen unterstreichen diese Ergebnisse die Spezifika der identifizierten *User Interface* Merkmale im Kontext von Videolernumgebungen, zum anderen zeigen sie Analogien und Gemeinsamkeiten zu ähnlich gelagerten Problemen und Lösungen auf. Interaction Design Patterns für Videolernumgebungen werden also durch ähnliche Pattern Languages kaum ansatzweise abgedeckt. Im folgenden Schritt des *Pattern Minings* geht es darum, die bisher gewonnen Erkenntnisse in schriftlicher Form als Patterns zu formulieren. Eine Komponente der Schriftform bilden „Verwandte Muster“. Neben Mustern innerhalb derselben Pattern Language zählen auch einige der in diesem Abschnitt identifizierten Beziehungen dazu.

### 4.2.4 Verfassen sowie Revision und Evaluation der Muster

Durch die Analyse und Klassifikation der Videolernumgebungen ist eine empirische Basis für mögliche Design Patterns entstanden. Diese so genannten Pattern Entwürfe wurden mit Design Patterns aus anderen Wissensdomänen abgeglichen und hinsichtlich möglicher Überschnei-

dungen untersucht. Diese Vorarbeiten bilden nun die Grundlage für die Verschriftlichung der Design Patterns. In diesem Teilabschnitt wird die iterative Vorgehensweise beim Schreiben und Überarbeiten der Muster erläutert. Neben den Elementen eines Musters liegt ein wesentliches Augenmerk auf der Revision und Evaluation der Musterbeschreibung mit Hilfe des *Shepherdings* und der *Writers' Workshops*. Diese Methoden sind insofern essentiell, um die Qualität der Patterns als wiederverwendbare, abstrakte Lösungen für häufig auftretende Probleme hinsichtlich der Gestaltung des *User Interfaces* von Videolernumgebungen zu sichern.

Das Verfassen der Muster gestaltet sich als iterativer Prozess. Die hier angewandte Vorgehensweise beim Schreiben der Patterns stützt sich im Wesentlichen auf die Ausführungen, die Wellhausen und Fiesser [2012] den Autoren von Pattern empfehlen. In Anlehnung daran sollten zunächst folgende Kontextfragen für jeden Pattern-Kandidaten beantwortet werden:

- Was ist das Besondere an der Lösung?
- Was ist enthalten und was ist ausgeschlossen?
- Wie kann man das Pattern von ähnlichen Ansätzen, Dingen oder Prozessen abgrenzen?
- Worin besteht der Aha-Effekt der Lösung?
- Was würde passieren, wenn man die Lösung nicht anwenden würde?
- Welche Beispiele sind bekannt?

Ein jedes Muster ist dabei gleich aufgebaut. Es entspricht in Textform, Struktur und Formatierung weitestgehend der Form wie sie Alexander et al. [1977] und andere Autoren von Patterns verwenden (z.B. Köppe [2012a]). Die Textform folgt einem narrativen Prosastil [Quibeldey-Cirkel 1999] mit einem Umfang je Pattern von maximal zwei Seiten inklusive der Abbildungen. Gegenüber einer schematischen oder tabellarischen Textform ist der narrative Stil besser lesbar, miteinander vergleichbar und aufgrund der starren Struktur und einheitlichen Formatierung ebenso einfach und schnell zu überblicken. Im Folgenden sind die einzelnen Strukturelemente mit Hinweisen zu ihrer Formatierung erläutert:

- **Name:** Der Name eines Musters gilt als Synonym für die darin beschriebene Lösung, weshalb eine kurze und prägnante Bezeichnung gewählt werden sollte.
- **Grad der Elaboration:** Hinter dem Namen des Musters können ein bis drei Sterne (\*) dargestellt sein, die angeben wie elaboriert ein Muster ist. Alexander [1979] wollte damit ausdrücken, wie wichtig die Autoren ein Muster erachten und wie überzeugt sie von der Lösung sind. Subjektive Einschätzungen dieser Art sind für jedoch schwer nachzuvollziehen. Aus diesem Grund wird ein quantifizierbares Maß aus der Inhaltsanalyse herangezogen: Die Anzahl der bekannten Realisierungen bildet die Grundlage zur Berechnung des Reifegrads. Darin kommen sowohl die Relevanz der Lösung angesichts ihrer Verbreitung, als auch die Überzeugung zum Ausdruck, eine vielfach erprobte Lösung zu wertschätzen. Operationalisiert werden hier die Terzile der analysierten Anwendungsfälle. Durch Terzile werden die nach ihrer Größe geordneten Häufigkeiten der Anwendungsfälle je Pattern in drei Abschnitte gleichen Umfangs geteilt. Je nachdem, ob die Anzahl der Fälle im unteren, mittleren und oberen Drittel liegt, werden ein, zwei oder drei Sterne hinter dem Namen des Musters angegeben. Die Berechnung der Terzile ist in Formel 4.3 dargestellt. Die

Häufigkeiten der Anwendungsfälle sind in geordneter Form in einer Rangliste  $x_1, x_2, \dots, x_n$  angegeben. Für die Terzile  $Q_{\frac{1}{3}}$  und  $Q_{\frac{2}{3}}$  nimmt  $p$  die Werte  $\frac{1}{3}$  bzw.  $\frac{2}{3}$  an.

$$\tilde{x}_p = \begin{cases} \frac{1}{2}(x_{n \cdot p} + x_{n \cdot p + 1}), & \text{wenn } n \cdot p \text{ ganzzahlig,} \\ x_{\lceil n \cdot p \rceil}, & \text{wenn } n \cdot p \text{ nicht ganzzahlig.} \end{cases} \quad (4.3)$$

Wobei  $z \in \mathbb{R}$  der Wert  $\lceil z \rceil$  die kleinste ganze Zahl ist, die größer oder gleich  $z$ .

- **Kontext:** Im ersten Absatz sind die Umgebungsbedingungen beschrieben, die unabhängig von der Lösung des Problems bestehen. Diesem Absatz folgen drei Karos (◆) als Abgrenzung zur Problembeschreibung.
- **Problem:** Im Abschnitt Problem wird in einem Satz erklärt, worin das Gestaltungsproblem besteht und was passieren würde, wenn es ungelöst bleibt. Zur besseren Lesbarkeit wird dieser Teil fett hervorgehoben.
- **Forces:** Unter Forces wird vermerkt, warum das Problem schwer zu lösen ist und warum andere Lösungen nicht in Betracht kommen. Idealerweise wird der Lösungsraum auf genau eine Lösung reduziert, indem alternative Lösungen durch Verweis auf andere Patterns entkräftet werden [Wellhausen und Fiesser 2012]. Diesem Abschnitt folgen abermals drei Karos (◆) als Abgrenzung zur Beschreibung der Lösung.
- **Lösung:** Im Abschnitt Lösung wird selbige in ein oder zwei Sätzen zusammengefasst und fett hervorgehoben. Die Lösung steht dabei im wechselseitigen Zusammenhang mit dem Problem. Wichtig ist eine Abstraktion von den aus der Inhaltsanalyse hervorgegangenen Einzelfällen hinsichtlich des Kerns einer Lösung. Im Gegenzug darf die Lösung auch nicht zu allgemein gehalten sein, sodass eine konkrete Umsetzung möglich wird. Bestandteil der Lösung sind auch Hinweise zur Umsetzung.
- **Konsequenzen:** Der Abschnitt Konsequenzen gibt Aufschluss darüber, was passiert, wenn die Lösung umgesetzt wird. Es werden positive und negative Konsequenzen angeführt. Dieses Element eines Musters beinhaltet auch eine Begründung der Lösung – hier durch Angabe einschlägiger Literatur aus den Bereichen Kognitionspsychologie, Lernpsychologie, Instruktionsdesign oder Usability. Darüber hinaus kommen in diesem Abschnitt auch mögliche Ausprägungen der Lösung zum Ausdruck.
- **Beispiele:** Durch kommentierte *screenshots* und Verweise auf externe Ressourcen werden dem Leser in diesem Abschnitt nicht nur anschauliche Beispiele der Lösung vorgestellt, sondern gleichzeitig Belege für die Existenz von (mindestens) drei Anwendungsfällen erbracht. Da die Patterns stets den Kern einer Lösung repräsentieren, sollen die Beispiele ein breites Spektrum an Umsetzungsmöglichkeiten aufzeigen. Die Quelle dieser Beispiele bilden die in der Inhaltsanalyse dokumentierten Informationsträger.
- **Verwandte Muster:** Wenngleich in den vorangegangenen Abschnitten bereits Verweise auf andere Patterns enthalten sein können, werden sie an dieser Stelle noch einmal zusammengefasst. Dies gilt im Übrigen auch für Verweise auf Muster in anderen Pattern Languages. Um beim Lesen Verweise auf andere Patterns schnell erfassen zu können, werden diese grundsätzlich durch Kapitälchen typografisch hervorgehoben (z.B.: ANNOTATED TIMELINE).

Entgegen dem Aufbau eines Patterns, folgt man beim Schreiben einer gesonderten Reihenfolge. Ausgangspunkt sind die aus den Anwendungsfällen hervorgegangenen und dem Autor somit vielfach bekannten Gestaltungslösungen. Aus diesen muss zunächst der Kern der Lösungen abstrahiert werden. Dabei ist es wichtig, die Lösung so zu beschreiben, dass sie dem Gestalter genügend Freiheitsgrade gewährt und vielfältigen Anforderungen gerecht werden kann, ohne dabei den Kern der Lösung zu verfehlen. Aus der Lösung erfolgt nun ein Rückschluss auf das ursächliche Problem.

Anschließend wendet man sich wieder dem Bereich der Lösungen zu und gibt die positiven und negativen Konsequenzen an, die mit der Anwendung der Lösung einhergehen. Eine jede Konsequenz ist einer Force zugeordnet, die im Anschluss abgeleitet werden. Forces drücken die mit dem Problem verbundenen Schwierigkeiten aus und schränken den Lösungsraum durch Angabe alternativer (Teil-)Probleme und Lösungen ein. Diese Einschränkung stellt auch eine Begründung der Lösung dar. Zum Schluss widmet man sich dem Kontext und gibt diejenigen Bedingungen und Anforderungen an, unter denen das Problem besteht und nach Anwendung der Lösung unveränderlich bleibt. [Wellhausen und Fiesser 2012]

**Revision und Evaluation** Auf Basis der ersten ausformulierten Muster kann die iterative Überarbeitung beginnen. Die Revision und Überarbeitung ist ein elementarer Prozess bei der Entwicklung von Design Patterns. Wird ein Pattern überarbeitet, wirken sich die Änderungen unter Umständen auch auf andere Patterns aus. Der Revisionsprozess kommt erst dann zur Ruhe, wenn ein oder mehrere Muster sich in ihrem Inhalt stabilisieren und sie durch die Rückmeldung anderer Personen ein Konsens bilden [Vlissides 1996]. Bei der Revision der vorliegenden Patterns kamen vier Methoden zur Anwendung: *Shepherdings*, *Writers' Workshops* sowie die Evaluation der Textverständlichkeit und die praktische Implementierung der Patterns in Lernumgebungen.

Zur Beantwortung der Frage, ob die Musterbeschreibung ausreicht, um die Lösung zu vermitteln, gibt es mehrere Lösungsansätze. Naheliegend ist die Begutachtung durch andere Fachexperten sowie solche, die es werden wollen und deshalb die Kollektionen von Design Patterns als Lernmaterial oder Nachschlagewerk nutzen. Ziel ist es also, den Nutzen und die Verständlichkeit der Musterbeschreibung für die Zielgruppe der Novizen auf dem Gebiet kritisch zu prüfen, um sie gegebenenfalls zu überarbeiten. In der Design Pattern Community haben sich mit dem *Shepherdings* und den *Writers' Workshops* zwei Methoden zur Begutachtung und Verbesserung von Mustern bewährt. Beide Methoden sind fester Bestandteil der PLoP-Konferenzen<sup>21</sup>, die seit 1995 an mehreren Orten der Welt jährlich stattfinden<sup>22</sup>.

Das *Shepherding* ist ein iterativer Revisionsprozess – ähnlich dem Peer Review Verfahren. Am Prozess beteiligt sind ein *Sheep* und ein *Shepherd*. Das *Sheep* hat ein oder mehrere Design Patterns zu einer PLoP-Konferenz als Teil eines Papers eingereicht. Als *Shepherd* fungiert in der Regel ein erfahrener Autor von Design Patterns. Seine Aufgaben besteht darin, dem *Sheep* Hinweise zur Verbesserung der Patterns und des Papers zu geben. Das *Sheep* kann diese Vorschläge aufgreifen oder mit dem *Shepherd* diskutieren. Noch bevor das Programmkomitee in

<sup>21</sup> PLoP steht für *Pattern Languages of Programs*

<sup>22</sup> Vgl. Aufstellung der PLoP-Konferenzen der Hillside Group: <http://hillside.net/conferences> (abgerufen am 15.02.2015).

Absprache mit dem *Shepherd* über die Annahme des Papiers entscheidet, durchlaufen *Shepherd* und *Sheep* in einem Zeitraum von sechs bis acht Wochen mindestens drei Feedback- und Überarbeitungsrunden. Dabei geht es weniger um eine fachliche Begutachtung der domänenspezifischen Problemstellungen und Lösungen, als um strukturelle, dem literarischen Genre der Design Patterns (vgl. Quibeldey-Cirkel [1999, S.57-59]) angemessene und vor allem konstruktive Hinweise zur Verbesserung der Design Pattern Beschreibung. Harrison [1999]

*Writers' Workshops* sind ein essentieller Bestandteil einer PLoP-Konferenz. Ein Workshop besteht aus mehreren Sessions, in denen jeweils ein Papier eines anwesenden Autors diskutiert wird. Wenngleich thematisch ähnliche Einreichungen in einem Workshop gruppiert werden, liegt auch hier der Schwerpunkt auf der Verbesserung von Struktur und Beschreibungsformat. Von den Workshopteilnehmern wird erwartet, dass sie alle zur Diskussion stehenden Pattern-Paper vor dem Beginn der Konferenz, zumindest jedoch vor der jeweiligen Session gründlich gelesen haben. In einer Session nimmt der Autor eines Papiers dann eine passive Rolle ein. Er wird lediglich gebeten, einen zentralen Abschnitt seiner Arbeit vorzulesen und einen Schwerpunkt für die anschließende Diskussion vorzuschlagen. Während der Diskussion wendet sich der Autor von der Gruppe ab und enthält sich der verbalen und nonverbalen Kommunikation („fly on the wall“). Die übrigen anwesenden Autoren sowie Gäste geben unter Leitung eines Moderator kritisches Feedback zu gelungenen und verbesserungswürdigen Passagen des Textes.

Neun Design Patterns, aufgeteilt in zwei Artikel mit fünf und vier Patterns, durchliefen im Rahmen der *European Conference on Pattern Languages of Programs 2014* (EuroPLoP) das *Shepherding* und zwei Sessions im *Writers' Workshop* (vgl. Seidel [2014b,e]). Während des *Shepherdings* erfolgten vier bzw. fünf Iterationen, während im *Writers' Workshop* zu Educational Patterns Rückmeldungen von fünf Personen in zwei einstündigen Sessions eingeholt wurden. Die Teilnehmer verfügten alle über einschlägige Lehrerfahrung, teilweise auch im Kontext von E-Learning sowie mit Lernvideos. Zehn weitere *Interaction Design Patterns* wurden im Januar 2015 zur EuroPLoP eingereicht und durchliefen das *Shepherding* sowie zwei Sessions mit acht Teilnehmern im *Writers' Workshops* (vgl. [Seidel 2015a,b]).

Um festzustellen, in wie weit die Musterbeschreibung auch Zielgruppen hilft, die wenig bis keine Expertise in der Entwicklung von Videolernumgebungen haben, wurde eine dritte Evaluationsmethode angewandt. Ziel war es, die Verständlichkeit der Kernbestandteile (Name, Kontext, Problem, Lösung) aller 45 Muster zu untersuchen. Vier Teilnehmer bekamen jeweils Texte der Kernbestandteile vorgelegt und sollten dazu mit Stift und Papier ein Konzept skizzieren, welches so aussagekräftig ist, dass ein Software Entwickler wissen würde, was zu implementieren sei. Die Texte enthielten absichtlich keine Beispielabbildungen oder Erläuterungen zur Umsetzung. Auf diese Weise waren die Teilnehmer angehalten, auf ihre Erfahrungen als Nutzer von Videolernumgebung zurückzugreifen oder für sie neuartige Gestaltungslösungen zu entwerfen. An der Untersuchung beteiligten sich drei Personen (eine Frau, zwei Männer), die als Hilfskräfte an der Professur für Produktionswirtschaft und Informationstechnik an der TU Dresden beschäftigt waren und sowohl privat, als auch im Rahmen ihres Studiums Lernvideos und dafür ausgelegte Lernumgebungen nutzen oder genutzt haben. Ein Teilnehmer hatte zudem grundlegende praktische Kenntnisse in der Entwicklung von Web-Anwendungen. Die von den Teilnehmern erstell-



ten Skizzen wurden vom Autor dieser Arbeit hinsichtlich ihrer Korrektheit und Vollständigkeit jeweils auf einer Skale von 1 bis 10 bewertet und bezüglich aufgetretener Missverständnisse untersucht. Auf Grundlage dieser Auswertung (vgl. Anhang A.10, S. 307) wurden Muster, die im Durchschnitt weniger als 8 Punkten in einer der beiden Dimensionen bekommen hatten, noch einmal überarbeitet.

Die Erprobung der Patterns in eigens implementierten Videolernumgebungen stellt die vierte Methode zur Evaluation und Revision der Patterns dar. Im Rahmen der Entwicklung eines generischen Software Frameworks für interaktive Videos (*VI-TWO*, siehe Kapitel 6, S. 215) wurden fast alle Patterns für eine vielseitige Verwendung implementiert. Daraus hervorgegangen sind bis dato sechs eigenständige Videolernumgebungen für verschiedene Auftraggeber, Zielgruppen und Untersuchungszwecke. Die Konzeption und Entwicklung förderte tiefgreifendes Verständnis für den Gestaltungsraum, die Gestaltungsprobleme und bestehende Freiheitsgrade in der Anwendung der Lösung. Insbesondere die Kombination von Mustern, d.h. das Zusammenspiel ließ sich nicht so umfänglich in der Inhaltsanalyse von Videolernumgebungen (siehe Abschnitt 6.3, S. 246) ermitteln. Zusätzlich konnten während des Einsatz der Lernumgebungen Prozessdaten erhoben und mit Methoden des Learning Analytics (z.B. Seidel [2014a]) ausgewertet werden (z.B. Seidel [2013, 2014c]). Ein Übersicht der kombinierten Patterns findet sich in Tab. 6.5 (S. 251).

Im Ergebnis wurden 45 Design Patterns für die Gestaltung des *User Interface* von Videolernumgebungen verfasst und mehrfach, teils aufgrund von Hinweisen aus der Forscher Community überarbeitet. Nachdem also die Muster im Einzelnen betrachtet wurden, bedarf es noch einer Gesamtbetrachtung, bei der die internen Beziehungen in Gestalt der Pattern Language im Mittelpunkt stehen.

#### 4.2.5 Konstruktion der Pattern Language

Die in den vorangegangenen Schritten aus den Pattern Entwürfen der Inhaltsanalyse in das Format der Design Patterns überführten und mit Hilfe von Schreibwerkstätten überarbeiteten Design Patterns sollen sich im letzten Schritt zu einer Pattern Language zusammenfügen. Die Konstruktion einer Mustersprache stellt das finale Ziel bei der Entwicklung von Design Patterns dar [van Welie und Veer 2003]. Eine Pattern Language entsteht grundsätzlich schon aus den Querverweisen zwischen den Mustern. Das aus den Querbeziehungen resultierende Netz ist ab einer gewissen Knotenzahl jedoch sehr unübersichtlich und schwer zu erfassen. Wie bereits in 4.1.2 (S. 103) erläutert, empfiehlt sich eine übergeordnete Klassifikation der Muster. Im vorliegenden Fall resultiert diese aus dem System von Dimensionen und Kategorien der Inhaltsanalyse (siehe S. 117) sowie der Unterscheidung zwischen Interaktionen mit einem oder mehreren Videos. Diese zweidimensionale Struktur wird nachfolgend in Abschnitt 4.3.1 auf S. 140 genauer erläutert.

An dieser Stelle ist es angebracht, noch einmal einen Schritt zurück zu gehen und die Vollständigkeit der Pattern Language kritisch zu hinterfragen.

**Vollständigkeit der Pattern Language** Wie lässt sich prüfen, ob eine Pattern Language alle Patterns enthält, die in Anbetracht der Gestaltungsaufgabe für die Zielgruppe relevant sind?



Van Welie und Veer wählten zur Evaluation ihrer Web Design Patterns einen Weg, um fehlende Patterns zu identifizieren:

„Therefore, one criterion could be that completeness is reached when the available patterns can account for all different qualitatively good designs one can find. In other words, when every ‘usable’ website out there can be described using a set of patterns, the language is complete. We say ‘usable websites’ because we are only interested in describing ‘good’ design. Alexander [et al.](1977) states that a pattern language is good when it is ‘morphological complete’. However, whenever new good sites start appearing the language may turn out incomplete again. Nonetheless, it gives us a practical method for mining.“ [van Welie und Veer 2003], siehe auch [Alexander 1979, S. 316-318])

Die hier angewandte Form der Inhaltsanalyse bekannter Videolernumgebungen folgte diesem Ansatz. Durch die Kodierung von Komponenten des *User Interfaces* wurden jedem Informationsträger Kategorien zugeordnet, mit deren Hilfe sich Lernumgebungen charakterisieren und beschreiben lassen. Während der Analyse wurden weitere Kategorien ergänzt, um die Gestaltungslösung möglichst vollständig durch Patterns beschreiben zu können. Der Umfang der aus den Analysekategorien hervorgegangenen Patterns ermöglicht die Beschreibung weiterer Videolernumgebungen mit Hilfe der Patterns. Wie detailliert sich Lernumgebungen beschreiben lassen, ist vom Detailgrad der in den Mustern festgehaltenen Lösungen abhängig. Muster auf einer abstrakteren Ebene, so Alexander [1979, S. 318], erfordern ausreichend untergeordnete Muster, die Lösungsvarianten konkretisieren. Eine Voraussetzung für eine Gliederung in untergeordnete Muster ist eine ausreichende Zahl an Praxisbeispielen. Im Fall von HYPERLINKS gelang es BRANCHING VIDEO und DETAIL ON DEMAND als eigenständige Varianten der Lösungen zu separieren. Auch für ANNOTATIONS konnten 12 Patterns zwei darunterliegenden Hierarchiestufen zugeordnet werden. Hinsichtlich der unterschiedlichen Ausprägungen des ASSESSMENTS, wie das von Kortenkamp und Blessing [2011] vorgestellte E-Learning Pattern VIDEO-CLIPQUEST, ist dies mangels weiterer Beispiele beispielsweise noch nicht gelungen.

Als ein Hilfsmittel zur Analyse einer Pattern Language empfehlen Hübscher et al. [2011] das *Universal Model of User Interface* [Baxley 2003]. Das Modell besteht aus neun Schichten (*Layer*), die den Prozess des *User Interfaces Designs* beschreiben (siehe Tab. 4.9). In einer Studie haben Hübscher et al. [2011] *Interaction Design Patterns* aus vier Pattern Languages ([Tidwell 2005; van Duyne et al. 2006; van Welie und Veer 2003; Yahoo! 2012]) auf Baxleys Modell abgebildet, um dadurch Schwerpunkte und Lücken hinsichtlich der Unterstützung beim Design von Benutzerschnittstellen zu identifizieren. Derselben Analyse wurden auch *Video Interaction Design Patterns* unterzogen. In Tab. 4.9 ist neben den Beschreibungen die Verteilung auf die neun Layer dargestellt. Details der Analyse mit den einzelnen Kategorisierungen sind im Anhang A.11 (S. 308) nachzulesen.

**Tabelle 4.9:** Verteilung der Patterns auf die neun Layer des *Universal Model of User Interface*

Layer	Beschreibung	Patterns	
1. <i>Conceptional Model</i>	Benutzer entwickeln aus der Erfahrung mit ähnlichen Anwendungen ein kohärentes Bild von der Funktionsweise und den grundlegenden Operationen einer Anwendung.	0	(0,0 %)
2. <i>Task Flow</i>	Der Aufgabenfluss beschreibt wie bestimmte Operationen zur Erreichung eines Ziels durchgeführt werden.	3	(6,7 %)
3. <i>Organisational Model</i>	Beschreibt wie Inhalte und Funktionalitäten innerhalb des Systems geordnet und strukturiert sind.	9	(20,0 %)
4. <i>Viewing and Navigation</i>	Benutzerinteraktionen, die sich auf die Darstellung auswirken und der Navigation dienen.	17	(37,7 %)
5. <i>Editing and Manipulation</i>	Benutzerverhalten, was permanente Änderungen der gespeicherten Information zur Folge hat.	10	(22,2 %)
6. <i>User Assistance</i>	Interaktionselemente, die den Anwender über Aktivitäten und Zustände des Systems informieren oder deren Benutzung erläutern.	3	(6,7 %)
7. <i>Layout</i>	Platzierung und Anordnung sichtbarer Bildelemente.	3	(6,7 %)
8. <i>Style</i>	Grafische Gestaltung der Benutzeroberfläche.	0	(0,0 %)
9. <i>Text</i>	Enthält alle geschriebenen, sprachabhängigen Elemente der Benutzerschnittstelle.	0	(0,0 %)
Summe		45	(100,0 %)

Der Schwerpunkt der hier behandelten Pattern Language liegt demnach auf den Layern (4) *Viewing and Navigation*, (5) *Editing und Manipulation* und (3) *Organisational Model*. Kein Pattern konnte den Layern *Conceptional Model*, *Style* und *Text* zugeordnet werden. Zur Identifikation von konzeptionellen und mentalen Modellen erscheint die Methode der Inhaltsanalyse als ungeeignet. Dennoch berücksichtigen einige Patterns konzeptionelle Modelle, die im Zusammenhang mit Video Playern und der videografischen Abbildung von Lehrveranstaltung bzw. Wissensvermittlung eine Rolle spielen, zum Beispiel:

- BASIC CONTROLS – Schaltelemente Play, Pause, Vor- und Zurück gehen nicht nur auf Videorekorder, sondern auch auf Tonbandgeräte zurück.
- BASIC CONTROLS / ANNOTATED TIMELINE – Horizontaler Verlauf von Zeitleisten in Anlehnung an die Leserichtung von rechts nach links sowie an Zeitstrahlen.
- SIMULTANEOUS MEDIA – Gleichzeitige Darstellung von Vortragenden und Präsentationsfolien analog zur räumlichen Anordnung von Person und Medien in einer Vorlesung.

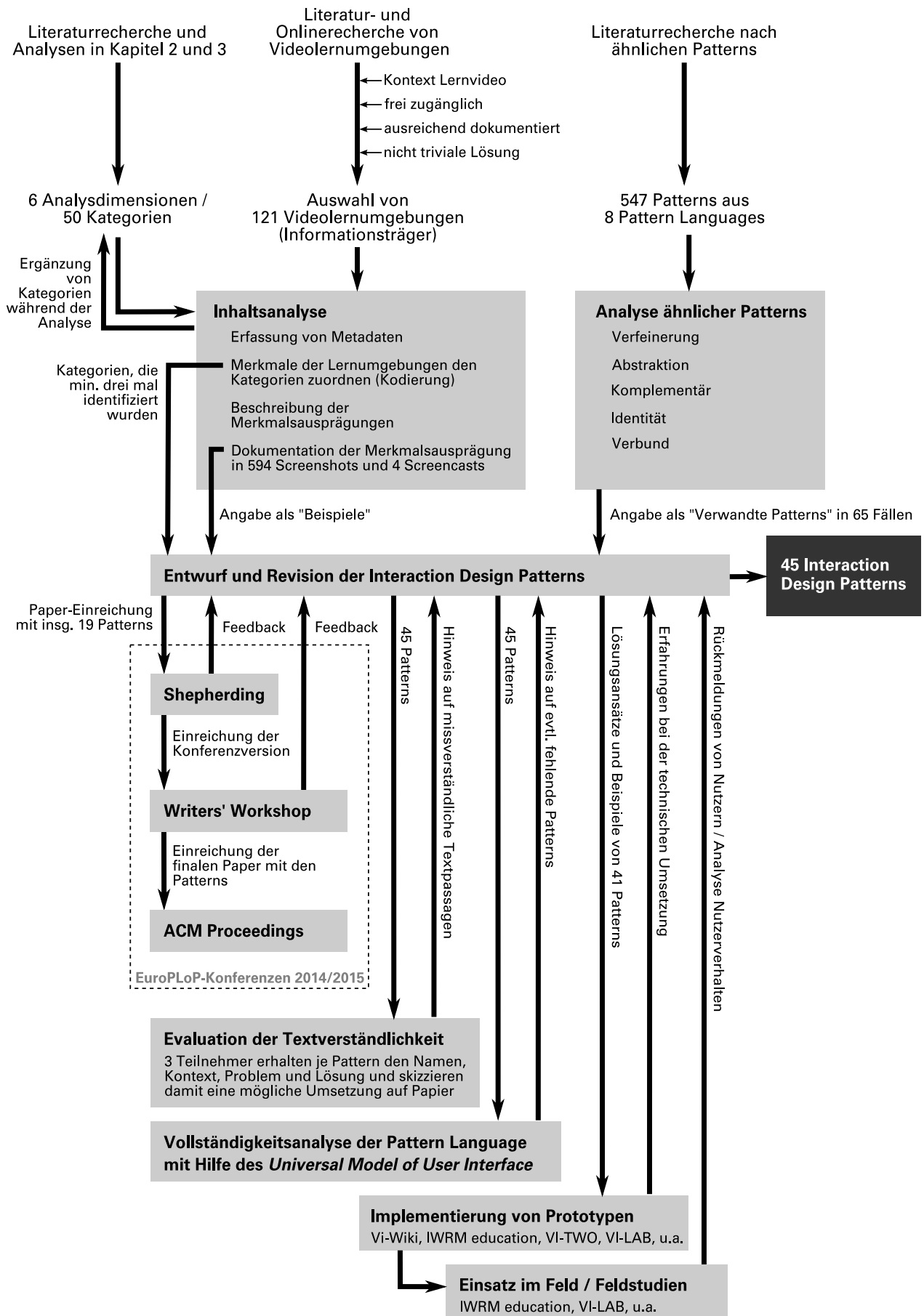
Die von van Welie und Veer [2003] benannten „Posture Pattern“ (im Onlinekatalog<sup>23</sup> als „Site types“ bezeichnet), wie z.B. Web-based Application, Portal Site oder E-commerce Site, wurden von Hübscher et al. [2011] als *Conceptional Model* eingestuft. *Posture Patterns* abstrahieren Lösungen für Anwendungen, die den gleichen Zweck erfüllen und deswegen ein ähnlich aufgebautes Seitenkonzept aufweisen [van Welie und Veer 2003]. Als spezifische *Conceptional Models* für Videoanwendungen lassen sich demnach einige der in Tab. 4.4 (S. 116) ausgewiesenen Gruppen der untersuchten Informationsträger auffassen. Folgende Gruppen wurden in vergleichbaren Pattern Languages (siehe Tab. 4.8) noch nicht als gesonderte Patterns berücksichtigt: Videolernumgebung, Portal, Autorenumgebung, Interaktiver Film, Annotationswerkzeug und Video Player.

Das *Layer Style* wurde nicht in eigenen Patterns thematisiert, da betreffende Lösungen an vielen Hochschulen, Unternehmen und anderen Organisationen oftmals durch *style guides* oder *Corporate Design Manuals* vorgegeben sind. Patterns zur Textgestaltung spielen im Kontext von Videos außerdem eine untergeordnete Rolle. Abgesehen davon enthalten weniger spezifische Design Pattern Languages bereits allgemeingültige Patterns (laut Hübscher et al. [2011] z.B. Tidwell [2005]; van Duyne et al. [2006]; van Welie und Veer [2003]). Die beiden letzten Layer Style und Text erscheinen dem Anwender zwar als besonders markant, jedoch sind sie für die Usability eines Systems weniger relevant [Baxley 2003].

Mit dieser Analyse kommt die in diesem Abschnitt vorgestellte systematische Elaboration von Design Patterns zum Abschluss. Die gesamte Vorgehensweise ist in Abb. 4.6 zusammenfassend dargestellt.

---

<sup>23</sup>Der Onlinekatalog von Martijn van Welie war unter <http://www.welie.com/patterns/> erreichbar, ist derzeit nur noch über die *Wayback Machine* des *Internet Archive* zugänglich: <https://web.archive.org/web/20140825181556/http://www.welie.com/patterns/index.php> (abgerufen am 15.02.2015).



**Abbildung 4.6:** Gesamtübersicht der Vorgehensweise zur systematischen Elaboration der Interaction Design Patterns für Videolernumgebungen

## Pattern Language für Videolernumgebungen

In diesem Abschnitt werden die Ergebnisse des zuvor beschriebenen *Pattern Minings* in Gestalt der Pattern Language für Videolernumgebungen vorgestellt. Im Zuge dessen wird nicht nur der innere Aufbau der Pattern Language erläutert, sondern auch ihre Beziehungen zu anderen Mustersprachen, die neben dem Interaction Design eine wichtige Rolle in der Konzeption und Entwicklung spielen. Damit nähert man sich auch der Frage, wie die Patterns anzuwenden sind und auf welche Weise sie bereits angewendet wurden. Das zentrale Werkzeug ist dabei die Wissensbasis, in der alle Patterns interaktiv aufbereitet sind. Eine repräsentative Auswahl der insgesamt 45 Patterns bilden den Abschluss dieses Teilkapitels. Alle übrigen Patterns erscheinen aus Platzgründen nur in einer Kurzdarstellung in Tab. 4.11 (S. 142).

### 4.3.1 Struktur der Pattern Language

Die Pattern Language umfasst insgesamt 45 Patterns. Diese Interaction Design Patterns beschreiben verbreitete Lösungen für wiederkehrende Probleme bei der Gestaltung und Entwicklung von Videolernumgebungen. Die Patterns sind entsprechend ihrer Gültigkeitsbereiche (vgl. Gamma et al. [1995, S. 14]) in zwei Schichten organisiert. Die erste Schicht enthält 27 Patterns, die die Mikrointeraktivität innerhalb des Videoplayers beschreiben. Darunter fallen alle Manipulationen und Benutzerinteraktionen, die sich auf die Darstellung innerhalb des Videos oder die Wiedergabe desselben beziehen [Riempp 2000; Schwan 2005; Zahn 2004]. In der zweiten Schicht werden 13 Patterns erläutert, die die Makrointeraktivität innerhalb der medialen Gesamtstruktur repräsentieren. Die Makrointeraktivität betrifft Manipulationen von einem oder mehreren Videos als Ganzes und innerhalb einer Kollektion mehrerer Videos (z.B. in einem Portal, einer Datenbank oder in einem CMS).

Unterhalb dieser beiden grundlegenden Schichten sind die Patterns hinsichtlich ihrer übergeordneten Gestaltungsziele kategorisiert, welche auch als eine Hierarchie von Gestaltungsproblemen (vgl. van Welie und Veer [2003] und Kunert [2009]) angesehen werden kann. Hierbei werden Patterns zusammengefasst, mit denen eine bestimmte Gruppe von Gestaltungslösungen verbunden ist. Diese stimmen mit den sechs Analysedimensionen aus Teilabschnitt 4.2.2 (S. 112) überein. In der folgenden Aufstellung steht die Anzahl an Patterns je Gruppe in Klammern dahinter.

1. *Grundfunktionen* (4)
2. *Zugang zu zeitbasierter Information* (13)
3. *Kontribution* (7)
4. *Strukturierung* (9)
5. *Selbstorganisation* (6)
6. *Layouts* (6)

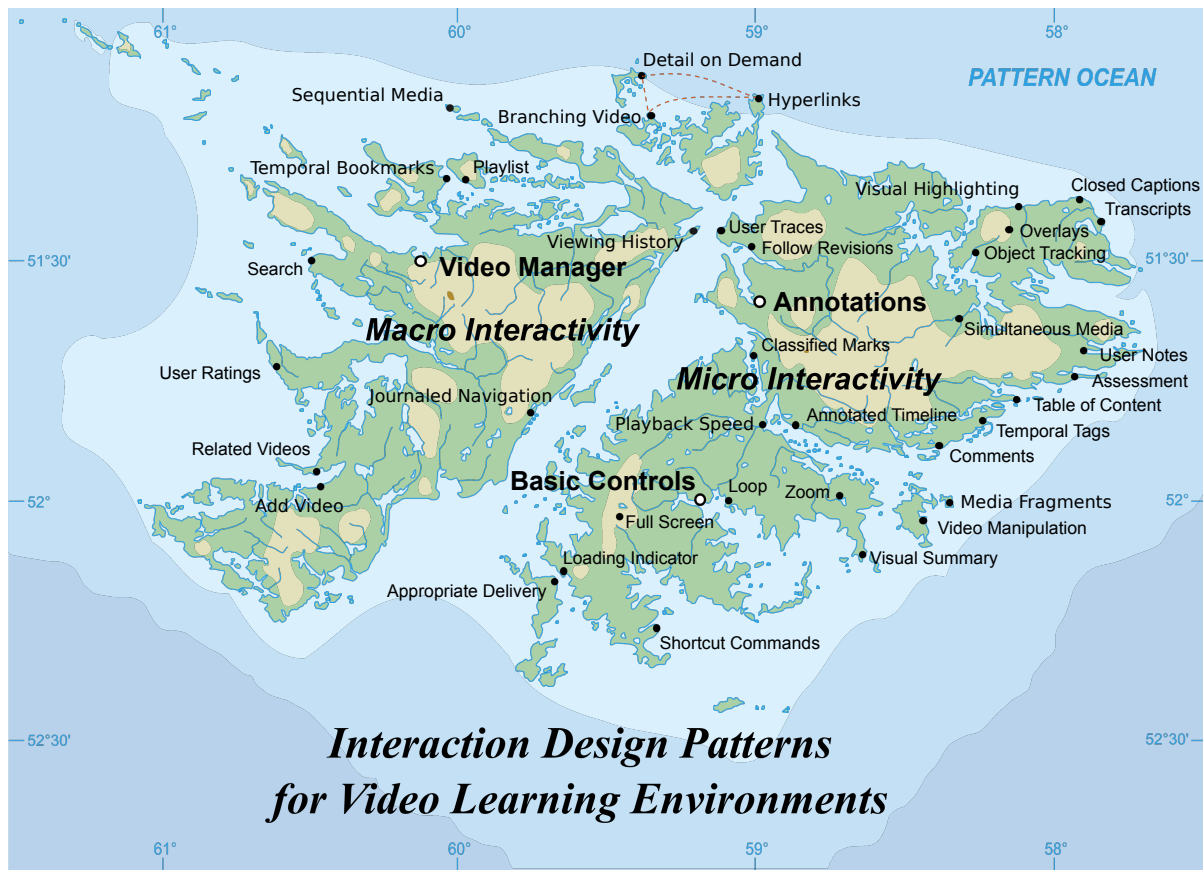
In Tab. 4.10 sind sowohl die beiden Schichten, als auch die funktionalen Gestaltungselemente in einer Matrix zusammengefasst. Eine alternative Darstellung der Pattern Language ist in Abb. 4.7 durch eine *Pattern Map* stilisiert. Auf der Karte sind drei Inseln zu sehen. Die auf den Inseln befindlichen Ortschaften sind nach Patterns benannt. Die Inselhauptstädte heißen BASIC CONTROLS, ANNOTATIONS und VIDEO MANAGER. Die beiden rechten Inseln liegen nah beiein-

**Tabelle 4.10:** Gliederung der Pattern Language in Mikro- und Makrointeraktivität sowie hinsichtlich funktionaler Aspekte.

	<b>Mikrointeraktivität</b>	<b>Makrointeraktivität</b>
<b>Grundfunktionen</b>	BASIC CONTROLS, APPROPRIATE DELIVERY, LOADING INDICATOR, SHORTCUT COMMANDS	
<b>Zeitbasierter Informationszugang</b>	TABLE OF CONTENT, TEMPORAL TAGS, PLAYBACK SPEED, ZOOM, VISUAL SUMMARY, ANNOTATED TIMELINE, TRANSCRIPT, CLOSED CAPTIONS, SKIP BACK, LOOP	SEARCH, TEMPORAL BOOKMARKS, JOURNALED NAVIGATION
<b>Kontribution</b>	ANNOTATIONS, COMMENTS, DIRECT AUTHORING, VIDEO MANIPULATION, REMIX, MULTI-TIMELINE EDITING	ADD VIDEO
<b>Strukturierung</b>	MEDIA FRAGMENTS, CLASSIFIED MARKS	VIDEO MANAGER, SEQUENTIAL MEDIA, RELATED VIDEOS, HYPERLINKS, BRANCHING VIDEOS, DETAIL ON DEMAND, USER RATINGS
<b>Selbstorganisation</b>	FOLLOW REVISIONS, USER NOTES, ASSESSMENT, USER TRACES	VIEWING HISTORY, PLAYLIST
<b>Layout</b>	FULL SCREEN, SIMULTANEOUS MEDIA, SYNCHRONIZED MAP, OVERLAY, VISUAL HIGHLIGHTING, OBJECT TRACKING	

ander und repräsentieren die Mikrointeraktivität, die große Insel rechts die Makrointeraktivität. Diese drei Inselhauptstädte subsumieren mehrere andere Muster und werden durch andere Muster ergänzt, die ohne sie gegenstandslos wären. Darüber hinaus sind für diese drei Muster vergleichsweise viele Beispiele (117/28/17) belegt. Muster, die mit BASIC CONTROLS in Beziehung stehen, betreffen die Darstellung und Steuerung audiovisueller Medien. Die mit Annotationen in Zusammenhang stehenden Muster decken Probleme und Lösungen ab, die bei der Informationsanreicherung des Videos entstehen (z.B. TEMPORAL TAGS oder Classified Marks) (vgl. Seidel [2015b]). VIDEO MANAGER, als drittes abstraktes Muster, repräsentiert die Notwendigkeit, mehr als ein Video handhaben zu können (vgl. Seidel [2014e]).

**Aufbau der einzelnen Patterns** Ein jedes Muster ist gleich aufgebaut. Die Grundbestandteile entsprechen der an Alexander et al. [1977] angelehnten und auf S. 131 beschriebenen Form. Die Bestandteile sind: Name, Elaborationsgrad, Kontext, Problem, Wechselwirkungen, Lösung, Konsequenzen, Beispiele und verwandte Muster.



**Abbildung 4.7:** Illustrative *Pattern Map*, auf der die Beziehungen zwischen den Mustern durch die Position auf der Karte bzw. auf den Inseln symbolisiert sind.

**Tabelle 4.11:** 45 Interaction Design Patterns für Videolernumgebungen

Design Pattern	Kurzbeschreibung
ADD VIDEO	Anwender können selbst Videos zur Lernumgebung hinzufügen.
ANNOTATED TIMELINE	Interaktive Abbildung zeitlicher Ereignisse auf der Zeitleiste
ANNOTATIONS	Ergänzung des Videos um zusätzliche Inhalte
APPROPRIATE DELIVERY	Auslieferung von Videos in angemessener Qualität und Bildgröße
ASSESSMENT	Im Video verankerte normative Lernerfolgskontrollen (Tests)
BASIC CONTROLS	Grundlegende Bedienelemente wie Play, Pause, Lautstärkeregler, Zeitleiste
BRANCHING VIDEOS	Hyperlinks im Video kennzeichnen Lernpfade.
CLASSIFIED MARKS	Bei einer großen Anzahl von Annotationen werden diese sichtbar klassifiziert.
CLOSED CAPTIONS	Sprache und sonstige Audioinformationen werden synchron zum Video als Untertitel dargestellt.
COMMENTS	Zeitgenaue Kommentare aus denen sich im Video verankerte Diskussionen entwickeln können.
DETAIL ON DEMAND	Details können bei Bedarf durch Verweise auf weitere Videos abgerufen werden.

(Fortsetzung auf der nächsten Seite)



(Tabelle 4.11 – Fortsetzung von der vorherigen Seite)

DIRECT AUTHORIZING	Anwender können unmittelbar während der Betrachtung ein Video bearbeiten oder anreichern.
FOLLOW REVISIONS	Änderungen am Video und den angereicherten Informationen sind in einer Änderungshistorie nachvollziehbar.
FULL SCREEN	Im Vollbildmodus sind nicht nur das Video, sondern alle relevanten zeitabhängigen Inhalte und Bedienelemente verfügbar.
HYPERLINKS	Zeitgenaue Verweise auf Ressourcen, die mittels zeitabhängiger und räumlich platzierter Anker im Video enthalten sind.
JOURNALED NAVIGATION	Beim Aufruf eines schon einmal betrachteten Videos wird die letzte Abspielposition aufgerufen.
LOADING INDICATOR	Rückmeldung an den Anwender, sofern die Medienwiedergabe gestört ist.
LOOP	Erlaube die mehrfache Wiederholung einzelner Ausschnitte oder eines ganzen Videos.
MEDIA FRAGMENTS	Gestatte die Wiederverwendung und Referenzierung von zeitlichen und räumlichen Videoausschnitten.
MULTI-TIMELINE EDITING	Zeitliche Ereignisse lassen auf einer mehrspurigen Zeitleiste bearbeiten, verschieben oder überblenden.
OBJECT TRACKING	Die Position von Annotationen ist mit sich bewegenden Objekten im Video verbunden.
OVERLAYS	Zeitweise werden zusätzliche Informationen in einem Layer über dem Videobild angezeigt.
PLAYBACK SPEED	Anpassbare Wiedergabegeschwindigkeit
PLAYLIST	Individuelle Auswahl von Videos für die spätere Betrachtung
RELATED VIDEOS	Angebot zusätzlicher, thematisch verwandter Lernvideos
REMIX	Bestehende Arrangements von Videos und zeitlichen Ereignissen bilden den Ausgangspunkt für individuelle Bearbeitungen.
SEARCH	Suche innerhalb einer oder in mehreren Videos
SEQUENTIAL MEDIA	Sequenziell verknüpfte Videos
SHORTCUT COMMANDS	Die wichtigsten Funktionen des Videoplayers sind auch durch bestimmte Gesten oder Tastaturkürzel ausführbar.
SIMULTANEOUS MEDIA	Unterschiedliche Medienströme werden bei der Produktion separiert und bei der Wiedergabe miteinander synchronisiert.
SKIP BACK	Gestattet es dem Anwender nur wenige Sekunden im Verlauf der Wiedergabe zurück zu springen.
SYNCHRONIZED MAP	Geografische Orte und Gebiete werden in Abhängigkeit der Wiedergabezeit neben dem Video angezeigt.
TABLE OF CONTENT	Inhaltsverzeichnis zum zeitgenauen Aufruf einzelner Abschnitte.
TEMPORAL BOOKMARKS	Sammlung von Verweisen, die eine zeitgenaue Adressierung von Abspielpositionen eines Videos ermöglichen.
TEMPORAL TAGS	Zeitgenaue semantische Annotation von Schlüsselwörtern in einem oder mehreren Videos
TRANSCRIPT	Stelle ein Transkript der Audioinformationen als ein Textäquivalent zum Video bereit. Abschnitte des Transkripts fungieren als Verweis auf die korrespondierende Szene im Video.

(Fortsetzung auf der nächsten Seite)

(Tabelle 4.11 – Fortsetzung von der vorherigen Seite)

USER NOTES	Gestatte Nutzern während der Wiedergabe schriftliche Notizen anzufertigen, die mit den Inhalten des Videos zeitlich verknüpft sind.
USER RATINGS	Anwender können Video und andere angereicherte Informationen bewerten.
USER TRACES	Stelle Nutzeraktivitäten bezüglich des Videos entlang der Zeitleiste dar.
VIDEO MANAGER	Aufstellung von Metadaten mehrerer, nach einem Kriterium ausgewählter Videos.
VIDEO MANIPULATION	Ermögliche grundlegende Funktionalitäten zur Bearbeitung der Videos.
VIEWING HISTORY	Darstellung der zuletzt betrachteten Videos
VISUAL HIGHLIGHTING	Zeitabhängiges Hervorheben von Objekten im Videobild
VISUAL SUMMARY	Zusammenfassung der visuellen Videoinhalte
ZOOM	Nutzer können visuelle Details im Bewegtbild vergrößert darstellen.

### 4.3.2 Angrenzende Mustersprachen

Für die Konzeption und Entwicklung einer Videolernumgebung sind neben den hier vorgestellten Interaction Design Patterns auch noch andere Pattern Languages relevant. In diesem Teilabschnitt werden daher Querbeziehungen zu ausgewählten Pattern Languages aus drei angrenzenden Disziplinen aufgezeigt. Entsprechend der Zielgruppen dieser Pattern Language werden Interaction Designer, Software Entwickler und Lehrende mit Pattern Languages aus ihren Fachbereichen angesprochen.

**Interaction Design** Die Gestaltung von Videolernumgebungen stellt einen Teilbereich des Interaction Designs dar. Videoplayer oder andere Funktionalitäten dieser Lernumgebungen sind oftmals nur Bestandteile der sie umgebenden LMS oder CMS. Für die Gestaltung und Entwicklung des *User Interfaces* dieser Umgebungen sollten allgemeinere Pattern Languages des Interaction Designs genutzt werden. Folgende Pattern Languages zählen dazu:

- Designing Interfaces: Patterns for effektive Interaction Design [Tidwell 2005, 2011]
- UI-Patterns<sup>24</sup>
- A Pattern Approach to Interaction Design [Borchers 2001]
- The Design of Sites: Patterns for Creating Winning Web Sites [van Duyne et al. 2006]
- Usability Patterns<sup>25</sup>
- Making Rich Web Application Architecture Usable<sup>26</sup>
- Patternry<sup>27</sup>
- Safety related patterns<sup>28</sup>
- Designing Social Interfaces Principles, Patterns, & Practices for Improving the User Experience [Crumlish und Malone 2009]

<sup>24</sup>Siehe <http://ui-patterns.com/patterns> (abgerufen am 21.04.2016).

<sup>25</sup>Siehe <http://www.sigchi.org/chi97/proceedings/short-talk/gca.htm#U6> (abgerufen am 21.04.2016).

<sup>26</sup>Siehe <http://www.sitepoint.com/architecture-usable/> (abgerufen am 21.04.2016).

<sup>27</sup>Siehe <http://patternry.com/patterns/> (abgerufen am 21.04.2016).

<sup>28</sup>Siehe <http://intuio.at/en/blog/?cat=patterns> (abgerufen am 21.04.2016).

**Software Entwicklung** Während die Interaction Design Patterns Gestaltungskonzepte vermitteln, enthalten sie noch keine Aussage darüber, wie eine gute Software Implementierung aussehen kann. Diese „low-level“ Lösungen sind vergleichbar mit den physikalischen Materialien und Strukturen in den Architektur Patterns von Alexander et al. [1977]. Das Pendant dazu bilden in der Software Entwicklung die Patterns der *Gang of Four* [Gamma et al. 1995], aber auch spezifischere Pattern Languages für Web-Applikationen, wie die „AJAX Design Patterns“. Darüber hinaus kommen auch Patterns aus der Entwicklung von Computerspielen in Betracht, die ähnlich wie CSCL-Scripts verschiedene Nutzerrollen, Level (Phasen), Ressourcen und Gruppenkonstellationen berücksichtigen. Folgende Pattern Languages wurden als relevant erachtet:

- Design Patterns: Elements of Reusable Object-Oriented Software [Gamma et al. 1995]
- AJAX Design Patterns<sup>29</sup>
- Game Programming Patterns<sup>30</sup>
- 400 Project<sup>31</sup>
- Patterns of Interaction<sup>32</sup>
- Where code and content meet. Design Patterns for Web Content Management and Delivery, Personalisation and User Participation. [Rüping 2009]

**Didaktik und E-Learning** Seit der Initiierung des *Pedagogical Pattern Project* im Jahre 1995 gibt es vielfältige Publikationen zu didaktischen Patterns (insb. [Bergin et al. 2012] und [Finlay et al. 2010]). In einem Teil davon geht es um E-Learning Patterns, die erprobte Lösungen für die organisatorische, didaktische und technische Konzeption und Durchführung elektronischer Lehrveranstaltung beschreiben. Darunter finden sich Patterns zu spezifischen Lernformen, wie dem kollaborativen Lernen [Briggs und de Vreede 2009; Goodyear et al. 2006; Pea und Lindgren 2008], aber auch solche zu einzelnen Tools wie E-Portfolios [Bauer und Baumgartner 2012] oder LMS [Avgeriou et al. 2003a,b]:

- Finlay et al. [2010]: The Pattern Language Network
- Bergin et al. [2012]: Pedagogical Patterns: Advice for Educators
- Bauer und Baumgartner [2012]: Electronic Portfolios in Higher Education
- Avgeriou et al. [2003a,b]: Patterns for Designing Learning Management Systems
- Köhne [2005]: Didaktischer Ansatz für das Blended Learning : Konzeption und Anwendung von Educational Patterns
- Pea und Lindgren [2008]: Collaboration Patterns in DIVER
- Briggs und de Vreede [2009]: *ThinkLets: building blocks for concerted collaboration*
- Goodyear et al. [2006]: Networked Learning

### 4.3.3 Repräsentation in einer Wissensbasis

Der beträchtliche Umfang der Patterns sowie die differenzierte Zielgruppe erfordern eine anwenderfreundliche Aufbereitung und Personalisierung der Pattern Language. Anwenden, die

<sup>29</sup>Siehe <http://ajaxpatterns.org/> (abgerufen am 21.04.2016).

<sup>30</sup>Siehe <http://gameprogrammingpatterns.com/> (abgerufen am 21.04.2016).

<sup>31</sup>Siehe <http://www.finitearts.com/Pages/400page.html> (abgerufen am 21.04.2016).

<sup>32</sup>Siehe <http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/patterns.html> (abgerufen am 21.04.2016).

nur über wenig Erfahrung in der Domäne verfügen, fällt es schwer aus mehr als zehn Patterns die für sie relevanten Patterns auszuwählen [Birukou 2010, S. 1]. Angesichts der 45 Interaction Design Patterns für Videolernumgebungen ist eine Darbietung in linearisierter Textform nicht anwenderfreundlich. Aus diesem Grund wurden die Patterns in eine anpassbare Wissensbasis integriert. Vergleichbare Repositorien für HCI-Patterns finden sich u.a. bei Lammi et al. [2014]; Tidwell [2005]; van Welie und Veer [2003]. Birukou [2010] hat in einer umfangreichen Studie die Such- und Selektionsfunktionen verschiedener Repositorien untersucht und kategorisiert.

**Tabelle 4.12:** Funktionen der Wissensbasis im Überblick

Funktion	Beschreibung
Browsing	Auflistung der Patterns mit der Option, diese dann im Detail zu betrachten.
Links	Navigation zwischen Patterns derselben oder einer anderen Pattern Language mittels Hyperlinks.
Kommentare	Anwender können ein jedes Element eines Patterns kommentieren und zusätzlich weitere Patterninstanzen oder Screenshots angeben. Alle Eingaben werden regelmäßig gesichtet und gegebenenfalls in der Patternbeschreibung oder der Datenbank berücksichtigt.
Wizzard	Durch mehrere Kontextfragen zu den Anforderungen an eine zu konzipierende Videolernumgebung kann der Anwender die Menge an Design Patterns auf eine für ihn relevante Auswahl reduzieren.
Vorauswahl	Vergleichbar mit einer Playlist können Anwender diejenigen Muster vormerken, die für sie von Interesse sind.
API	Für weitere Analysen oder die Integration in andere Anwendungen lassen sich die Patterns in einer strukturierten Form als JSON und im Format der <i>Pattern Language Markup Language</i> (siehe Anhang A.14, S. 318) abrufen.
Suche	Innerhalb der Muster sowie in einzelnen Komponenten der Patterns ist eine Volltextsuche möglich (z.B. in den Forces oder Beispielen).

Die Wissensbasis hierfür bildet ein eigenentwickeltes CMS zur Verwaltung von Design Patterns. In dem System sind alle Musterbeschreibungen in Bild und Text hypermedial dargestellt. Die Struktur der Mustersammlung kommt durch Hyperlinks sowie mehrere Übersichten zum Ausdruck. Die Beschreibungen enthalten Verweise auf exemplarische Musterinstanzen im WWW sowie ähnliche Muster in anderen Sammlungen. Bestandteil dieses Angebots ist auch eine Datenbank der in Abschnitt 4.2.2 (S. 112) untersuchten Videolernumgebungen. Weitere Funktionen der Wissensbasis sind in Tab. 4.12 aufgelistet. Die Wissensbasis, wie auch ihr Quellcode inklusive der Datenbank sind online frei verfügbar unter <http://designing-video-interfaces.nise81.com/> bzw. <https://github.com/nise/designing-video-interfaces> (jeweils abgerufen am 15.02.2015).

**Entscheidungsmatrix und Wizzard** Die Implementierung aller Patterns in einer Lernumgebung entspricht dem „maximal“ Möglichen. Ein Überangebot an Funktionen kann einen Anwender jedoch überfordern. In der Praxis genügt deshalb oftmals schon eine Teilmenge an Patterns, um die Anforderungen an einen spezifischen Anwendungskontext zu erfüllen. Um die Suche und Auswahl relevanter Patterns zu unterstützen, wurde ein Entscheidungsmodell in die Wissensbasis integriert. Vergleichbare Ansätze haben Pearson und Shen [2010, S. 6-7] mit

einem regelbasierten System und Greene et al. [2003, S. 2-3] mit benutzerdefinierten Regeln auf Grundlage von Kontextfragen und „Design Pattern Selection Rules“ realisiert.

Das im Rahmen dieser Arbeit entwickelte Modell basiert auf einer boolschen Entscheidungsmatrix, in der 12 Kriterien des beabsichtigten Anwendungskontexts den Patterns gegenübergestellt sind. Die Kriterien adressieren das Arrangement an Lernmedien (Medienart, Videoanzahl, Relationen), die Eigenschaften der Videos (Dauer, Detailgehalt / Dynamik) sowie die vorgesehene Lernform (selbstreguliert, kollaborativ). Ein Kriterium kann entweder zutreffen oder nicht zutreffen. Ob es für ein Pattern zutrifft, hängt vom Kontext-Statement des Patterns ab. Beispielsweise wird im Kontext-Statement des Patterns *Playlist* auf die eigenständige Organisation von mehreren Videos verwiesen, woraus sich im Modell eine Zuordnung zu Kriterium *F2* (mehrere Videos) und *F9* (selbstständiges Lernen) ergibt. Die Zuordnungen sind durch Wahrheitswerte (0 oder 1) repräsentiert und ermöglichen somit aussagelogische Disjunktionen der folgenden Art:

$$Playlist \Leftarrow F2 \vee F9 \quad (4.4)$$

In einigen Fällen bestehen auch Abhängigkeiten zwischen zwei Kriterien, die durch die Repräsentation in boolscher Algebra aufgelöst werden können. Im Anhang A.12, S. 310 ist das vollständige Modell in einer Tabelle dargestellt.

Ähnlich wie bei de Moel und van der Veer [2011] ist das Modell innerhalb der Wissensbasis an einen Fragen-Wizzard gekoppelt. Jede Frage bezieht sich auf ein Kriterium. Den Nutzern der Pattern Language steht es frei, die folgenden 12 Fragen zu beantworten.

- F1* Beinhaltet die Lernumgebung mehrere Medien (z.B. Videos, Bilder, Texte / Hypertexte)?
- F2* Sind mehrere Videos in die Anwendung integriert?
- F3* Stellen Sie weitere Medien synchron zum Video dar?
- F4* Umfasst Ihre Zielgruppe Personen mit eingeschränkten visuellen, auditiven oder sprachlichen Fähigkeiten (z.B. Personen mit Sehschwäche, Schwerhörige oder Nicht-Muttersprachler)?
- F5* Ist es erforderlich innerhalb der Videos oder zwischen Videos und anderen Lernressourcen Verknüpfungen herzustellen?
- F6* Ist mindestens ein Video länger als 10 Minuten?
- F7* Besitzt mindestens ein Video eine hohe visuelle Dynamik oder enthält es viele visuelle Details?
- F8* Ist die Menge der verfügbaren Lernressourcen einschließlich der Videos festgesetzt oder durch die Lernenden erweiterbar?
- F9* Sollen sich die Lernenden selbstständig mit den Lernressourcen auseinandersetzen?
- F10* Beabsichtigen Sie die Schaffung eines Rückkanals für Kommentare oder Anmerkungen?
- F11* Interagieren die Lernenden innerhalb der Anwendung miteinander oder bearbeiten sie eine gemeinsame Aufgabe?
- F12* Gehen Sie davon aus, dass Ihre Zielgruppe vornehmlich mobile Endgeräte mit kleinerem Display nutzt?

Im Anfangszustand sind lediglich die vier Patterns vorausgewählt, die als Grundelemente unabhängig vom medialen, didaktischen und technischen Kontext einer Anwendung zu berücksichtigen sind. Sobald eine Frage im Wizzard mit „ja“ beantwortet wird, erhöht sich die Anzahl an Patterns. Wird eine Frage ausgelassen oder mit „nein“ beantwortet bleibt die Menge konstant. Die Reihenfolge der Beantwortung wirkt sich nicht auf die Ergebnismenge aus.

Anzumerken ist hierbei, dass das Ergebnis lediglich eine Entscheidungshilfe darstellt. Alle vorgeschlagenen Patterns sollten hinsichtlich ihrer Passgenauigkeit zur didaktischen Zielstellung einzeln geprüft werden.

#### 4.3.4 Anwendungs- und Einsatzszenarien

In den zurückliegenden Abschnitten wurde der Aufbau und die Zusammensetzung der Pattern Language erläutert und ihre Repräsentation als Text und hypermediale Wissensbasis vorgestellt. Die in der Wissensbasis implementierte Entscheidungsmatrix deutet bereits eine Möglichkeit zur Anwendung der Patterns an, die sich auf die Konzeption und Realisierung von Videolernumgebungen bezieht. Es wurde bislang jedoch noch nicht viel darüber gesagt, wo und vor allem wie die Patterns darüber hinaus angewendet werden können. In diesem Teilabschnitt stehen drei Anwendungsszenarien im Mittelpunkt, in denen die Patterns bislang zum Einsatz gekommen sind:

- Entwicklung eines Software Frameworks zur Implementierung der Patterns
- Analyse von bestehenden oder in Entwicklung befindlichen Videolernumgebungen
- Wissenstransfer mit Hilfe von Design Patterns

**Konzeption und Entwicklung eines Software Frameworks für interaktive Videos** Laut Calvo und Turani [2010] bestehen E-Learning Frameworks aus Design Patterns und Software Komponenten. Der in diesem Anwendungsszenario verfolgte Ansatz trägt dieser vereinfachten Formel Rechnung, indem ein Framework für Videolernumgebungen mit Hilfe der hier vorgestellten Design Patterns konzipiert und entwickelt wurde. Die Design Patterns repräsentieren dabei Funktionalitäten, die sich in vielen Videolernumgebungen wiederfinden und deshalb auch in einem Framework enthalten sein sollten, was Entwickler dazu befähigt neue Lernumgebungen zu entwickeln. Das entstandene Framework und seine Architektur ist Gegenstand von Abschnitt 6.1 (S. 216). Es besteht im Kern aus einem Videoplayer, der um einzelne *Widgets* erweitert werden kann. Jedes Pattern auf Mikro-Ebene wurde dabei auf ein *Widget* abgebildet. Aus den in den Mustern enthaltenen *Beispielen*, den genannten *Forces* und den *Consequences* der Lösungsanwendung, konnten Determinanten für eine flexible Anpassung der Software Komponenten bestimmt werden. Somit können die *Widgets* mittels *CSS* in ihrer äußeren Gestalt und anhand von Parametern hinsichtlich ihrer Funktion flexibel angepasst werden. Für jeden Parameter ist dabei ein überschreibbarer Standardwert definiert. Angesichts dieser einfachen Konfiguration sollen auch Interface Designer bzw. Personen ohne dezidierte Programmierkenntnisse in die Lage versetzt werden, eine Videolernumgebung gemäß ihren Vorstellungen und Anforderungen zu definieren.



**Evaluation und Verbesserung bestehender Videolernumgebungen** Innerhalb einer Domäne eröffnen Sammlungen von Design Patterns eine einfache Möglichkeit, um bestehende Anwendungen hinsichtlich ihrer funktionalen Gestaltung zu analysieren [Björk et al. 2003; van Welie und Klaasse 2004]. Van Welie und Klaasse haben auf diese Weise Museumswebseiten anhand von Interaction Design Patterns von [van Welie und Veer 2003] untersucht [van Welie und Klaasse 2004]. Björk et al. [2003] empfehlen ein leicht abgewandeltes Vorgehen zur Analyse von Spielen. Eine Kombination dieser beiden Vorgehensweisen wurde hier exemplarisch zur Analyse der Videolernumgebung *eKomMed*<sup>33</sup> angewandt. *eKomMed* dient Studierenden der Medizin dazu, sich auf Arzt-Patienten-Gespräche in ihrer späteren beruflichen Praxis mit Hilfe von Videos vorzubereiten. Die Lernenden sind dabei angehalten, die in den Videos dargestellten Gesprächssituationen einzuschätzen und gemeinsam zu diskutieren. Bei der Analyse geht man folgendermaßen vor:

1. *Kontext ermitteln*: Ausgangspunkt bilden die Kontextvariablen, die eine Videolernumgebung charakterisieren. Diese werden mit Hilfe des Entscheidungsmodells (siehe 4.3.3) ermittelt.
2. *Abgleich mit Lernumgebung*: Für jede Unterseite bzw. jedes Seitenelement wird geprüft, ob die darin zum Ausdruck kommenden Gestaltungslösungen einem der in Schritt 1 ermittelten Design Patterns entsprechen. Sich aufzeigende Abweichungen werden diskutiert. Darüber hinaus gehende Gestaltungslösungen werden notiert und in der Pattern Language identifiziert. Falls dies nicht gelingt, kann es sich bei der Lösung um einen Kandidaten für ein neues Pattern handeln.
3. *Verbesserungsvorschläge*: In Abhängigkeit des Videomaterials, technischer Limitationen und der didaktischen Zielstellung wird empfohlen,
  - die laut Schritt 1 noch nicht implementierten Patterns anzuwenden,
  - die Notwendigkeit der darüber hinaus instantiierten Patterns zu prüfen und
  - für instantiierte Muster Gestaltungsalternativen mit Hilfe der in den Patterns beschriebenen Lösungen und Beispiele in Betracht zu ziehen.

Die am Beispiel von *eKomMed* ermittelten Analyseergebnisse der Videolernumgebung sind im Anhang A.13 auf S. 311 im Detail dargestellt. Um jedoch eine umfassende Analyse einer Lernumgebung zu erzielen, sollten auch andere Instrumente, wie z.B. die erweiterten Usability Guidelines (siehe Anhang A.2, S. 261), einbezogen werden.

**Wissenstransfer mit Hilfe von Design Patterns** Design Patterns sind vielfach Gegenstand der Lehre. Dies gilt insbesondere im Software Engineering (z.B. Köppe [2012b]; Muller et al. [2007]), aber auch im Bereich der Mensch Computer Interaktion gibt es Belege in der Literatur (z.B. Borchers [2002]; Chatzigeorgiou et al. [2008]; Koukouletsos et al. [2009]). Dabei kann man drei Vorgehensweisen zur Vermittlung von Design Patterns unterscheiden [Griffiths und Pemberton 1998]: (1) die theoretische Betrachtung von Design Patterns in ihren Anwendungsdomänen; (2) die praktische Anwendung von Patterns im Gestaltungsprozess (z.B. in der Software Entwicklung [Chatzigeorgiou et al. 2008; Muller et al. 2007]) und (3) die Identifikation von Patterns durch Studierende (z.B. Borchers [2002]; Köppe [2012b]). Angesichts der vorlie-

<sup>33</sup>Siehe <http://www.ekommed.de/> (abgerufen am 7.12.2015). Das Angebot ist aus urheberrechtlichen Gründen nur für autorisierte Nutzer zugänglich.



genden Pattern Language interessiert uns an dieser Stelle der zweite Ansatz, den Griffiths und Pemberton [1998] als „teaching *through* Pattern Language“ bezeichnet. Angewendet wurde dieser Ansatz in einem hochschulübergreifenden mediendidaktischen Angebot mit dem Titel „Facharbeitskreis »Videos in der Lehre«“<sup>34</sup>.

Der Facharbeitskreis bietet Lehrenden sächsischer Hochschulen ein offenes Forum zur Vorstellung und Diskussion videobasierter Lehr-Lern-Formate und unterstützender E-Learning-Technologien. Am 26. Juli 2014 wurde der Arbeitskreis an der TU Dresden gegründet und kam seitdem vier weitere Male in Präsenz und einmal per Telefonkonferenz zusammen. Organisiert und moderiert wurden die Treffen und Aktivitäten vom Autor dieser Schrift. Die Teilnehmer verfügen in der Regel über ein heterogenes Vorwissen. Sie unterscheiden sich hinsichtlich ihrer Lehrerfahrung, der Erfahrung mit dem Medieneinsatz in der Lehre sowie ihrem mediendidaktischen und medientechnischen Wissen voneinander. Bezüglich der Konzeption und Entwicklung von Videolernumgebungen handelt es sich bei den Teilnehmern um Novizen und beim Moderator um einen Experten.

Die Arbeitsweise des Facharbeitskreises erlaubt allen Teilnehmern, ihre jüngsten Erfahrungen und Erkenntnisse im Videoeinsatz in der Lehre vorzustellen sowie gezielte Fragen und Probleme in die Diskussion einzubringen. Neben didaktischen Konzepten für Lehrveranstaltungen werden auch Konzepte und Prototypen von Videolernumgebungen oder produzierte Lernvideos besprochen. Design Patterns für Videolernumgebungen wurden dabei nicht gesondert thematisiert, jedoch implizit in den Diskussionen über angemessene E-Learning-Werkzeuge und eigenentwickelte Lernumgebungen angesprochen. Ziel der Wissensvermittlung anhand von Design Patterns ist es, den Teilnehmern erprobte Lösungen für den Einsatz ihrer Videos in der Lehre zu vermitteln. Der Ansatz zur Wissensvermittlung anhand von Design Patterns orientiert sich dabei an der Vorgehensweise beim Ausformulieren von Patterns [Wellhausen und Fiesser 2012] und ist durch Muller et al. [2007] inspiriert. Ausgangspunkt sind fehlende Patterns sowie richtig oder falsch angewendete Patterns.

Der Moderator weist die Teilnehmer in einer von ihnen vorgestellten Videolernumgebung auf ein Pattern hin und leitet die Identifikation der einzelnen Pattern-Komponenten durch gezielte Fragestellungen an. Die Rolle des Moderators besteht darin, Fragen zu stellen und die Diskussion zusammenzufassen. Durch die strukturierte Diskussion sollen die Teilnehmer lernen (ihre) Gestaltungslösungen bzw. -varianten kritisch zu hinterfragen, indem sie Notwendigkeiten, Vorteile und Nachteile sowie von Variationen und Alternativen im Kontext eines (eigenen) Anwendungsfalls differenzieren.

Der folgende schematische Ablauf wurde in der Sitzung des Facharbeitskreises am 29. September 2014 mit sieben Teilnehmern an der TU Chemnitz anhand der Pattern OPEN CAPTIONS, TABLE OF CONTENT, OVERLAYS, SIMULTANEOUS MEDIA und VIDEO MANAGER erprobt<sup>35</sup>.

<sup>34</sup>Siehe Webseite <http://elearning.ihl-zittau.de/fak/videos-in-der-lehre/> (abgerufen am 16.05.2016).

<sup>35</sup>Siehe Protokoll <http://elearning.ihl-zittau.de/fak/videos-in-der-lehre/protokoll-fak-videos-in-der-lehre-26-09-2014/> (abgerufen am 15.02.2015).

1. *Lösung*: Der Moderator identifiziert ein fehlendes respektive ein richtig oder falsch angewendetes Design Pattern innerhalb der durch andere Teilnehmer vorgestellten Anwendungen, indem er auf die Lösung aufmerksam macht.
2. *Problem*: Alle Teilnehmer werden gebeten, die Beweggründe zur Umsetzung dieser Lösung zu erörtern: „Warum bedarf es dieser Lösung? Worin besteht die Notwendigkeit? Wie würde die Anwendung ohne diese Lösung funktionieren?“
3. *Konsequenzen*: Als nächstes folgt die Diskussion der Konsequenzen, die mit der Lösung einhergehen: „Welche Vor- und Nachteile bringt diese Lösung? Muss die Lösung genau so aussehen oder gibt es weitere Varianten?“
4. *Forces*: Um den Spielraum für komplementäre oder synergetische Gestaltungslösungen einzuschränken, werden Alternativen zur Lösung gesucht: „Welche alternativen oder ergänzenden Lösungen gibt es?“
5. *Kontext*: „Lässt sich diese Lösung verallgemeinern oder gilt sie nur unter bestimmten Bedingungen?“

Im Nachgang an das Treffen erhalten die Teilnehmer die besprochenen Design Patterns als PDF-Dokument. In den Patterns sollten sich die wesentlichen Aspekte der vorangegangenen Diskussion wiederfinden.

#### 4.3.5 Exemplarische Interaction Design Patterns

Für diesen Abschnitt wurden sechs exemplarische Interaction Design Patterns ausgewählt (siehe Tab. 4.13), die einerseits die Interaktion auf Makro- und Mikroebene und die andererseits alle sechs funktionalen Gruppen repräsentieren. Alle Muster durchliefen das *Shepherding* und einen *Writers' Workshop* im Rahmen der EuroPLoP 2014 oder 2015. Die vollständige Darstellung der Design Patterns findet sich unter der URL <http://designing-video-interfaces.nise81.com> (abgerufen am 18.07.2016).

**Tabelle 4.13:** Übersicht der ausgewählten Design Patterns

Name	Interaktivitätslevel	Funktionale Gruppe	EuroPLoP
SHORTCUT COMMANDS	Mikro	Grundfunktion	2015
PLAYBACK SPEED	Mikro	Zeitbasierter Informationszugang	2014
ANNOTATIONS	Mikro	Kontribution	2015
DETAIL ON DEMAND	Makro	Strukturierung	2014
PLAYLIST	Makro	Selbstorganisation	2014
VISUAL HIGHLIGHTING	Mikro	Layout	2015

## Shortcut Commands \*\*

Video players incorporate different designs and provide different user interfaces.



**Graphical user interfaces are not accessible for the blind without a screen reader. These interfaces can also become frustrating for users who are not impaired in their vision, especially for frequently executed tasks.**











According to Shneiderman und Plaisant [2004], it is crucial to consider the needs of diverse users from „novices to experts, age ranges, disabilities, and technological diversity“ [Shneiderman und Plaisant 2004]. Users who are blind, who have low vision or hand tremors find it difficult or impossible to make use of a video player that only provides a graphical user interface [Caldwell et al. 2008]. One reason for that is the non-standardized user interface facilitating BASIC CONTROLS as well as additional controls (e.g. ZOOM or SKIP BACK).



**Enable keyboard shortcuts and touch gestures for the most frequent user interactions. Specifically, define those shortcuts that most of the video players have in common.**

Many video players implement a set of core keyboard commands and touch gestures to provide BASIC CONTROLS. Especially blind people benefit from gestures since the graphical user interface is not accessible for them without a screen reader [Kane et al. 2011]. The shortcuts in Table 4.14 could be identified in many different video players (see example), including *YouTube* as the most common one.

**Table 4.14:** Typical commands and the assigned keyboard shortcuts and touch gestures.

Command	Keyboard shortcut	Touch gesture
play/pause the video		double tap
skip forward		drag to the right
skip backward		drag to the left
skip to 10% ... 90% of the playback time	 ... 	–
skip to beginning		–
volume up		drag up
volume down		drag down
toggle full screen		–
exit full screen mode or video		–

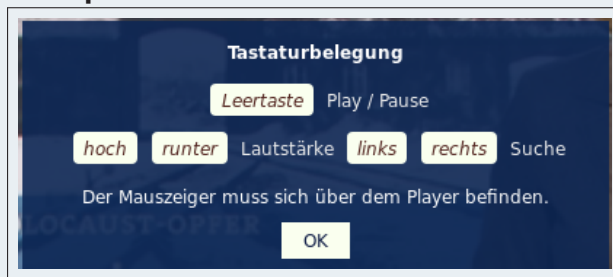
It is remarkable that the reading direction from left-to-right has become a universal standard for video players. Both the play button and the timeline are oriented to the right.

However, it is a challenging task to apply the recommendations for keyboard shortcuts and touch gestures on highly interactive video players. An ANNOTATED TIMELINE as well as an OVERLAYS or other types of ANNOTATIONS require complex interactions that are difficult to map on shortcuts.

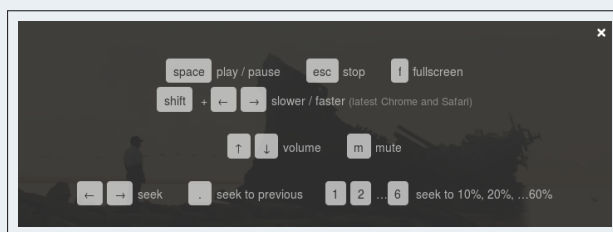
In particular, it is challenging to make keyboard shortcuts platform-independent, e.g. for different web browsers and operating systems. Further conflicts can arise through the use of single key press commands. Especially if the user is required to enter text (e.g. USER NOTES, COMMENTS or ASSESSMENT) or the same key is already mapped as a common user-agent keyboard command.

Other interactive elements should be made accessible by the use of Tab and Enter only. Coding recommendations on how to do that are provided by the W3C [Caldwell et al. 2008; W3C 2011c] Although the mentioned shortcuts might be very common, they need to be communicated to the user, e.g. on a help page.

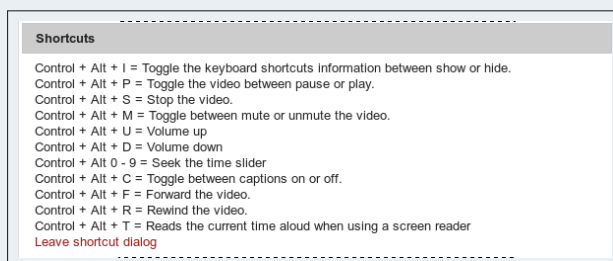
## Examples



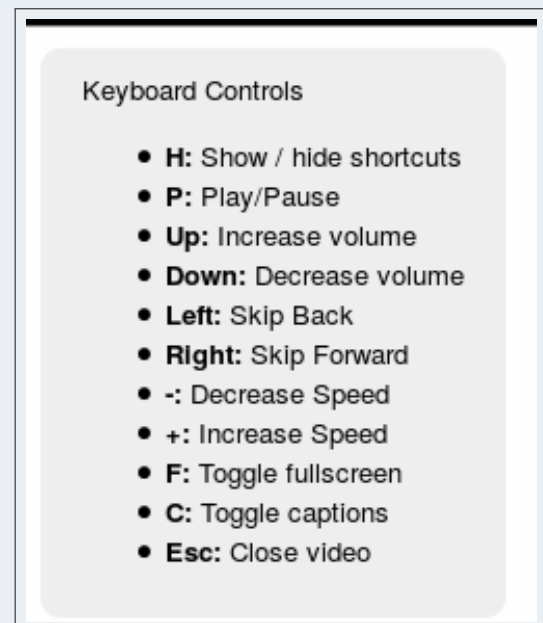
**Figure 4.8:** The news cast *Tagesschau* is utilizing the *Projekktor* player, which supports common keyboard commands. See <http://www.projekktor.com/> (last accessed 2015/11/30).



**Figure 4.9:** The video player that is used for the interactive documentary *CloudsOverCuba* covers the most common shortcuts, including those to adjust playback speed. See <http://cloudsovercuba.com/> (last accessed 2015/11/30).



**Figure 4.10:** *Lernfunk / Opencast Matterhorn* is a negative example because it requires the user to learn unusual keyboard short cuts. See <http://www.opencast.org/matterhorn> (last accessed 2015/11/30).



**Figure 4.11:** *Coursera* additionally introduces short cuts to control the playback speed. See <https://www.coursera.org/> (last accessed 2015/11/30).

**Related Patterns** BASIC CONTROLS, see also KEYBOARD ONLY in [Tidwell 2005, p. 17] and CHOOSING THE RIGHT KEYS in [Kunert 2009, p. 122].

## Playback Speed \*\*

The presentation of video contents can result in the two extremes of a cognitive load. On the one hand, overload can be caused by fast scene changes or volatile details in visual dynamic processes. Mental underload, on the other hand, is related to almost unchanging images or irrelevant content.



**The density of presented information in a video differs from the expected level of detail, sometimes it is perceived as too slow, sometimes as too fast.**

Although a timeline slider (BASIC CONTROLS and ANNOTATED TIMELINE) could be used to fast forward or backward it is difficult to keep a constant frame rate, particularly for short durations.

The range in which the playback can be slowed down and speeded up is bound to the human ability of perception. The lower bound represents almost still images as far as the video compression and capturing frame rate allows it. The upper bound represents overspeeded image sequences those details, e.g. gestures and voice intonation are not noticeable any more.



**Integrate a mechanism to change the speed of presentation from slow motion to normal speed to time-lapse and vice versa.**

If the user needs to follow critical steps the video can be slowed down. This is also useful if the user wants to take notes (USER NOTES) or add ANNOTATIONS during the playback. On the other hand instructions can be speeded up to get an overview or to move faster through sections of known content. Also trailers and screen credits can be processed faster. Nevertheless the timeline slider always corresponds to the playback speed by showing the movement within the temporal dimension.

Instead of a fixed sequence and pace of information presentation the viewer's mental activities are less constrained by the author's choice on how to design the video [Schwan und Riempp 2004]. Beside the benefit of slow motion and time lapse playback an adjustable playback speed enables visual and audible browsing and searching. By fast forward playback the user can get an overview of the visual contents and proceed until a scene of interest appears.

In case the video is played within the range of normal and double playback rate it is even possible to listen and comprehend speech. Fast listening is an advantage over VISUAL SUMMARY techniques that aggregate key visual content as still images.

SIMULTANEOUS MEDIA such as slides or pictures as well as OVERLAYS and other time-depended contents are effected by changes of the playback rate. These elements can be dedicated for user interactions but become inaccessible at high playback rates.

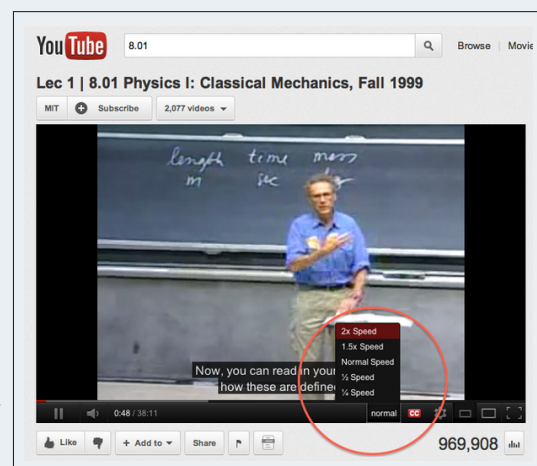
There is no common user interface metaphor representing playback speed controls. The pitching wheels as circular representation known from software DVD players can hardly be handled by correlating mouse or touch gestures. Sliders on the other hand are already stressed as timeline and volume control.

## Examples

- Schwan und Riempp [2004] pointed out the cognitive benefits of interactive control over the playback speed in a user study. The participants learned from an instructional video how to tie nautical nodes.
- A scientific prototype of Goldman [2007] allows to go forward and backward in time when dragging the objects in the corresponding direction<sup>36</sup>
- At *Coursera* the presentation pace can be controlled by the + and - keys from 0.3 to double speed.
- Since the time when VCR became popular, double arrows on the remote control were used to change the playback speed if pressed repeatedly.
- *YouTube* offers, in the United States of America, five options to adjust the playback to 1/4, 1/2, Normal, 1.5x and 2x speed.
- Modern web browsers provide an interface to adjust the *playbackRate* of HTML5 videos.



**Figure 4.12:** *MySpeed* is an extension for web browsers that allows the control of the presentation pace of Adobe Flash™ videos in the range of 0.3 and 3 [Enounce 2014].



**Figure 4.14:** At *YouTube* five discrete steps to adjust the playback speed can be selected.



**Figure 4.13:** *Cyber-Link* DVD player software *PowerDVD* provides a wheel control around the play button to play videos up to two times slower and 16 times faster.

## Related Patterns

BASIC CONTROLS, SKIP BACK, LOOP, and VISUAL SUMMARY.

<sup>36</sup>See a demo at: <http://vimeo.com/2345579?pg=embed&sec=2345579>, last viewed: June 1st, 2014.



**Annotations \* \* \***

A learning environment is not closed, but rather open for contributions made by teachers and students.



**Compared to text or images, it is difficult to modify a video file. Videos are technically self-contained, which implies their use as a broadcast medium in terms of teaching and learning. The involved people are impeded from making contributions or additions that are directly related to the time-based information structure.**

Videos are comparatively large media files that need to be re-encoded after every editing session. A video can be considered as a self-contained media element that can include all other symbol systems like images, text, sound, and speech [Wetzel et al. 1994]. Video cannot be changed without using tools that are capable of re-encoding the results into a suitable video format. Hence, a typical editing process requires much time, memory, and computational power. The result is an irreversible change of the original content. Furthermore, the resulting content is not comprised by SEARCH.

One way to deal with that is to split up large files in order to get small files that are easier and faster to edit. Small units can be lined up and extended with other material in terms of SEQUENTIAL MEDIA. In an extreme case an audio file synchronized with an animated stream of different content types would remain as a type of SIMULTANEOUS MEDIA.



**Make video extensible for the user through spatial and temporal annotations. Place them on top or beside the video.**

If important information is added to a resource or to certain parts of it, we consider the information as annotation. In terms of video, an annotation is anchored as an event in the temporal layout. An event can be related to a point in time or a time range within the temporal layout. According to the spatial extent video annotations can also correspond to a spatial position or space in a layer on top of the video. If an annotation is anchored in both time and space we refer to it as a spatio-temporal annotation.

Adding layers of annotations to a video does not affect the video itself and therefore makes it possible to reuse it.

However, there are several ways to extend a video. Basically the spatial and temporal dimensions are extensible by adding SIMULTANEOUS MEDIA such as presentation slides. The video frame itself can be augmented with further information layers to accomplish OVERLAYS such as VISUAL HIGHLIGHTING or OPEN CAPTIONS.

Another purpose of annotations is the addition of navigational aids like HYPERLINKS, TABLE OF CONTENT or TEMPORAL TAGS. Hence, it supports the searchability by relating semantics with time-related content.

Annotations are further useful to communicate and discuss content-related issues with other users (e.g. COMMENTS). In educational settings annotations have been used to analyze video-captured situations like sports trainers at work (e.g. [Vohle und Reinmann 2014]) or healthcare-related communication between doctors and patients.

Annotations are not restricted to a certain type of media. Although text might be most commonly images, audio, video (ADD CONTENT), or complex contents like web pages or maps are also feasible.

While access rights for video playback are granted by default editing requires a more sophisticated rights management. For some people it might be permissible to add certain types of annotations that others might be allowed to read, edit, or delete. This can go along with a shift between the roles of the producer and consumer of instructional resources. Quality and quantity of the contributions may have to be monitored or moderated.

Certain types of annotations are typically added by the instructor as part of the content preparation. For instance, a teacher anchors presentation slides (SIMULTANEOUS MEDIA) to a previously captured talk and defines a TABLE OF CONTENTS as well as a short quiz (ASSESSMENTS) every few minutes of the playback time.

Problems can arise when a video gets heavily annotated. In the case of too many OVERLAYS, the video itself fades into the background. Even when further information is presented simultaneously, a split attention effect can cause cognitive overload [Clark et al. 2006]. Beside that it becomes difficult to overview the temporal structure of the annotated content. Therefore an ANNOTATED TIMELINE could be helpful if annotations are distinguished by certain criteria in the way of CLASSIFIED MARKS. From the chronological order it is not obvious who has added what annotation at what time. Therefore a possibility to FOLLOW REVISIONS can help to unfold the editing process in order to comprehend or participate.

**Related Patterns** SIMULTANEOUS MEDIA and SYNCHRONIZED MAP specify annotations beside the video while OVERLAYS such as VISUAL HIGHLIGHTING or OPEN CAPTIONS specify solutions for displaying additional information inside the video frame.

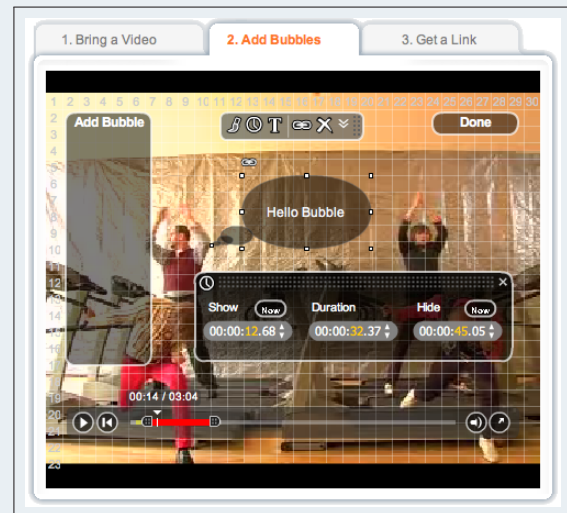
HYPERLINKS, TEMPORAL TAGS, and TABLE OF CONTENT draw attention to navigational aspects of annotations. COMMENTS shed light on the communication and analysis that is temporally and spatially anchored in the video.

See also SHARED ANNOTATION [Schümmer und Lukosch 2007, S. 219].

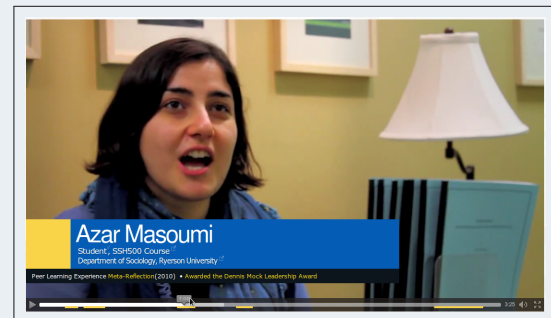
## Examples



**Figure 4.15:** The *MediaWiki* Extension *Vi-Wiki* enables a collaborative way to annotate spatio-temporal hyperlinks to videos inside a wiki page as well as in the creole markup editor [Seidel 2012b]. See <https://github.com/nise/vi-wiki> (last accessed 2015/11/30).



**Figure 4.16:** Annotate speech bubbles and links in an authoring environment called *BubblePly*.



**Figure 4.17:** Annotated captions implemented by the use of *popcorn.js*. See <http://ryersonssh500.com/> (last accessed 2015/11/30).

**Detail on Demand \*\***

Transfer of knowledge requires particular prior knowledge. The prior knowledge of individual learners almost always differs to some extent.

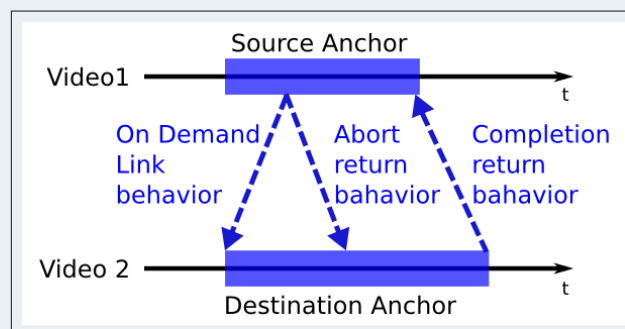


**Within an instructional video you can neither expect the same prior knowledge from every learner, nor can you explain every mentioned term or concept in detail to consider individual prior knowledge.**

The gap between expected and existing prior knowledge is often addressed by providing additional content. Additional learning resources can be offered constantly in parallel or at the end of the video. In both cases the resources are timely unrelated with the specific content that they ought to complement. In parallel presented content sometimes appears to be distracting because it anticipates material that is presented later on in the video. The other way around linking the content at the end comes too late in order to give access to necessary prior knowledge. Therefore, only additional content in the narrowed sense of the word should be provided as RELATED VIDEOS. Nevertheless, it has to be considered that learners can lose track of the originated video when accessing any linked resource from inside the video. That amplifies if the linked resource offers further links of that kind.



**Provide details on demand by referring to sequences of videos after whose presentation the original video continuous.**



**Figure 4.18:** The concept of detail on demand with abort and completion return behaviours regarding Girgensohn et al. [2003].

In contrast to usual **HYPERLINKS** leading to another website or a position in time, this kind of link is dedicated to a certain sequence in a target video. If a user follows a detail on demand link the source video will halt while the target sequence starts playing. The playback can be aborted by following a return link back to the origin. If the playback is not aborted by the user it will automatically return to the origin when the dedicated sequence is completed (see Figure 4.19). This pattern is related with the *cycle* hypertext pattern that Bernstein [1998] identified in hyperfiction documents. Detail on demand gives the user the opportunity to comprehend

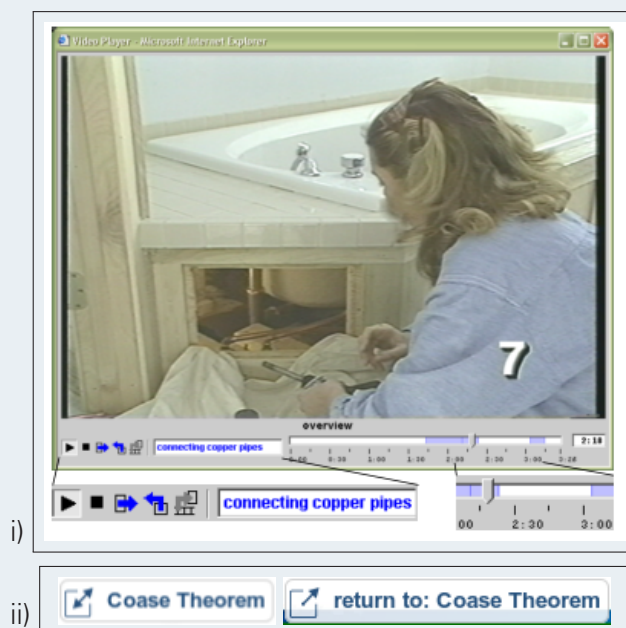
further details that are necessary to understand the originated content. Such details could be a definition of a term or an explanation of a concept. While dealing with that details the surrounding context of the initial video is only one click away. Watching video sequences on demand is like zooming into a picture. The incorporated return behaviour to the initial video is like zooming out to get back to the whole picture again.

Learners should not lose the indicated learning path when following a HYPERLINK to another video. The behaviour of automatically returning to the original video let the learners keep track. Therefore the linked sequences should not be too long. Nevertheless, links to additional videos can be offered as RELATED VIDEOS.

Video-based details that have been requested on demand can also contain further links to on-demand content. Although loops can be avoided, a multilevel recursive decline and ascent, through aborting or terminating a sequence, makes it difficult to orient towards the original video. To avoid cognitive overload, the hierarchy of linked videos should not be beyond one or two levels. A JOURNAL NAVIGATION can help to go back and forth in the navigation history. A bread crumb menu as Tidwell [2005] suggests could provide guidance for not getting lost in hyperspace.

### Examples

Girgensohn et al. [2003] introduced details on the demand in the player and authoring environment *Hyper Hitchcock* (see Figure 5). Later Leidel et al. [2012] applied that concept to a hypervideo network of 43 eLectures.



**Figure 4.19:** i) In the Hyper Hitchcock player Hyperlinks are indicated on the timeline and labelled with a description next to it. Only one link at a time is provided whereas a video called on demand just offers a link to return to the origin. Girgensohn et al. [2003]. ii) Source and destination link anchors at *IWRM education*<sup>5</sup> that symbolise the availability of a detail (left) and the link back to originated content (right).

**Related Patterns:**

HYPERLINKS, RELATED VIDEOS. JOURNALED NAVIGATION. See also DETAILS ON DEMAND [van Welie und Veer 2003], EXTRAS ON DEMAND [Tidwell 2005, p. 45-46], and Hypertext CYCLES and JOYCE'S CYCLE [Bernstein 1998].

---

<sup>5</sup>See <http://www.iwrm-education.org/>, last viewed on June 1st, 2014.

**Playlist \*\***

Videos in a large collection are filed in a predefined category system.



**When browsing a large collection of videos, users need to organize resources by their own needs. Selected, previously watched resources should not need to be searched over and over again.**

The need for a personal organisation of learning activities and content can be derived from the number of videos and the amount of playback time. The more resources are available the more possible learning paths can be followed and the greater is the demand to organize it.

A personal organisation of learning resources incorporates different activities, e.g.:

- Making a preselection of items when searching and browsing (SEARCH, VIDEO MANAGER).
- Sorting or grouping resources of interest in a personal classification system (e.g. by topic).
- Storing of organized content, at least for the time of use.
- Sharing of the arranged content with peers.

Unlike algorithmically assorted RELATED VIDEOS the user's own choice of videos is a relevant demand.

In contrast to TEMPORAL BOOKMARKS, the organized items refer to videos as a whole instead of specific parts of it. Apart from that, bookmarks are not necessarily stored within the learning environment.



**Provide learners a playlist, where they can collect and store subsets of a video collection for later use. These personal subsets can be freely extended or reduced by videos that are part of the learning environment.**

For the ease of access, the user should to some extent be allowed to set his or her own content structure on top of the given one.

The concept of playlists is known from music software or other media players. Also browser bookmarks can be considered as a user-generated list of playable items.

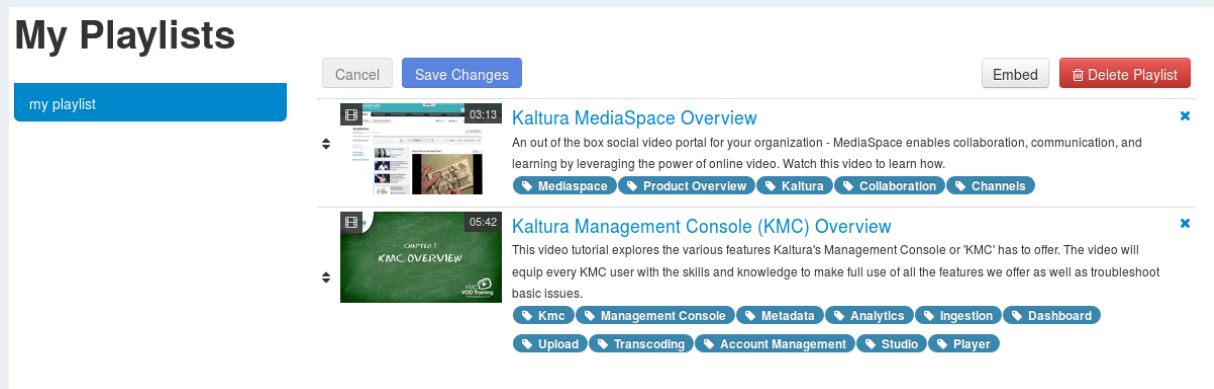
Playlists help to organize personal learning activities within a large amount of audiovisual resources. But that does not exclude their use within a group of learners, no matter if the list is publicly accessible or not. Organizing a playlist outside the learning environment by collecting TEMPORAL BOOKMARKS is not an added value feature of the environment itself.

The more items a playlist contains the more hierarchical or semantic structuring is recommended. That can be achieved through folders and subfolders (e.g. multiple playlists) as well as through personal notes or tags. Finally the possibility to bring the items in a user-defined order can be considered as a basic way to bring the items into a linear order.

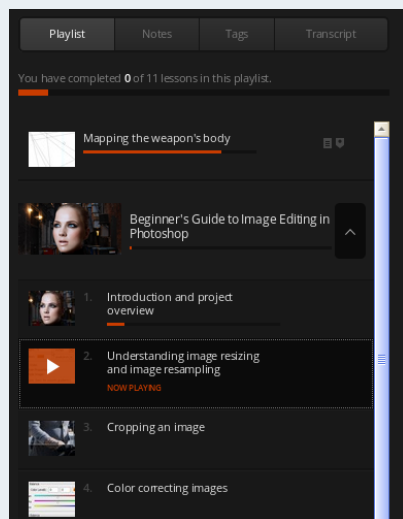


Playlist do not need to be stored persistently. When visitors do not need to register, playlist items can be stored until the end of the session.

### Examples



**Figure 4.20:** At *Kaltura*<sup>5</sup> the items in a playlist can be added, sorted and removed in a VIDEO MANAGER. It is also possible to embed the whole playlist into another website.



**Figure 4.21:** At *digital-tutors.com* users can collect and order videos and lessons containing multiple videos in a playlist. Furthermore videos can be collected separately in a “backpack” for later download in order to watch them offline.

### Related Patterns

TEMPORAL BOOKMARKS, BRWOSING HISTORY, SEARCH, USER TRACES, and RELATED VIDEOS. See also COLLECTOR [van Welie und Veer 2003] and SAVING [Crumlish und Malone 2009].

<sup>5</sup>See for further details.

## Visual Highlighting \*

A narrated video includes multiple visual aspects or details that are relevant to understand the content.



### **Learners are not aware of a visually presented object that is referred to by the narration or online discussion.**

If a visual object gains attention by the narrator its spatial position should be described if it is not clearly distinguishable from other objects presented at the same time. It can be difficult for the user to identify the currently relevant visual information. As a consequence comprehension and learning are negatively affected because audio information cannot be matched with its corresponding visual sections.

A relation between audio and the corresponding visual content could be considered during the production or post-production of the video as well as in later online discussion or analysis. Whether a spatial position is described by the speaker or visually etched in the original video content, it will only express the intention of the instructor or producer. Learners may draw their attention to different aspects when writing a COMMENT or defining ANNOTATIONS.



### **Highlight visual objects or details inside the video as long as they are necessary to understand the oral presentation or content-related communication.**

The purpose of highlighting visual elements excludes interactions that are inherent to OVERLAYS in general and HYPERLINKS in particular.

Visual highlighting can be derived from the *Signalling Principle* of the Theory of Multimedia Learning by Mayer [2009]. Highlighting or marking draws attention to essential visual aspects of the presentation. It supports mental selection and organization during the perception of instructional videos. Ideally there should be only one element highlighted at a time. Therefore it can be seen as an analogy to real-life lessons where teachers point at objects with their fingers, a stick, or a laser pointer. It is an advantage if a part of a moving image calls somebody's attention while its context stays visible. The relationship between the parts and the whole carries a meaning beyond the spatial arrangement. ZOOMING is an alternative way to draw attention but it neglects this whole/part relationship.

This method can also be applied in discussions among students who want to focus on spatial objects rather than on explaining their location verbally. Consequently instructors and students can make use of the possibility to highlight content elements.

There are many possible ways to pay attention to an object in a video frame without manipulating the video itself. Technically it needs to be realized with a visual layer on top of the video. Colored frames or boxes can be put around the object in question, arrows can point at them, or the surrounding area can be grayed out. Highlighted elements should always appear distin-

guishable from their background. In a few cases it might be necessary to reinforce the visibility by blinking colors or animating appearing or disappearing overlays.

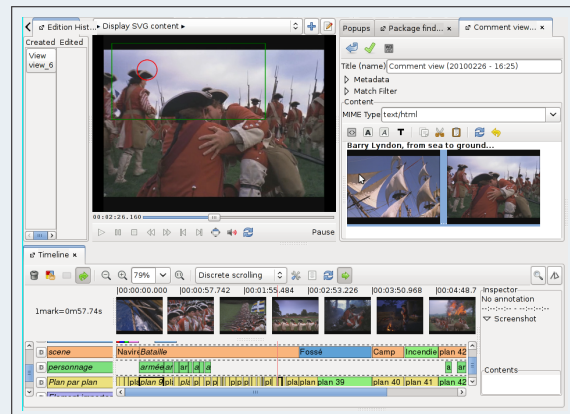
If the highlighting is related to a moving object it becomes necessary to change its position accordingly. OBJECT TRACKING algorithms can handle this to some extent. Annotation by hand becomes cumbersome if an object moves in a non-linear manner.

However, highlighting certain objects interrupts the visual representation as a whole. In the same way as elements emerge from the background, the elements in the background do get less attention. That's why it is important to give the user enough time for viewing the whole picture before focusing upon a detail.

## Examples



**Figure 4.22:** In *eduBreak* sequences of movements of captured sport scenes are emphasized through colored boxes. Students are encouraged to comment and highlight visual aspects in order to discuss these with their peers [Vohle 2010]. See <http://www.edubreak.de/> (last accessed 2015/11/30).



**Figure 4.23:** *Advene* features the highlighting of moving visual objects through object tracking. See <http://liris.cnrs.fr/advene/> (last accessed 2015/11/30).



**Figure 4.24:** In *DIVER Panoramic* (2002) parts of the 360°panorama video can be marked to be watched in detail. The underlying concept is called „guided noticing“. It is a part of process that consists of pointing to, marking out or highlighting a specific aspect. These aspects can be named and categorized. See <http://diver.stanford.edu/guided2.html> (last accessed 2015/11/30).

**Related Patterns** ANNOTATION, OVERLAY [Kunert 2009, p. 109] See also TELEPOINTER [Schümmer und Lukosch 2007, S. 359]

## Zusammenfassung

Das Ziel dieses Kapitels bestand darin, bewährte Gestaltungslösungen des *User Interfaces* von Videolernumgebungen aus einer Vielzahl von Einzelfällen zu identifizieren und in Form von Design Patterns ihre Notwendigkeit durch Angabe der dafür ursächlichen Probleme im jeweiligen technischen oder didaktischen Kontext zu begründen. Die Zielsetzung adressierte Forschungsfrage FF 1 nach bewährten Lösungen für häufig auftretende Probleme (Design Patterns) hinsichtlich der Gestaltung des *User Interfaces* von Videolernumgebungen.

Im zurückliegenden Kapitel wurde eine Methode zur Identifikation von Interaction Design Patterns vorgestellt, die auf einer Inhaltsanalyse von individuellen *User Interface* Lösungen und der Analyse von ähnlichen Patterns aus ähnlichen Mustersprachen beruht.

Das Ergebnisse der Inhaltsanalyse von 121 Videolernumgebungen hat gezeigt, dass die meisten Anwendungen (104) über weniger als 10 Funktionsmerkmale verfügen und gut die Hälfte der untersuchten Merkmale nur in einer relativ kleinen Anzahl von Anwendungen (24) realisiert sind. Damit wird deutlich, dass es keine universellen Videolernumgebungen gibt und je nach didaktischer Zielstellung ganz unterschiedliche Anwendungen in Betracht kommen. Als Pattern-Kandidaten wurden 45 der 50 Untersuchungskategorien bestimmt, da sie in mindestens drei Anwendungen unabhängig voneinander realisiert worden sind. Im Rahmen dieser Analyse konnten sehr viele Beispiele für Umsetzungsmöglichkeiten der in den Patterns zu beschreibenden Lösungen gesammelt und als Screenshots (459) oder Videos (4) dokumentiert werden. Um Dopplungen mit Mustern thematisch ähnlicher HCI-Patterns zu vermeiden und stattdessen Anknüpfungspunkte herzustellen, wurden acht thematisch verwandte Pattern Languages aus dem HCI-Bereich untersucht. Nur 12 % der 547 betrachteten Patterns konnten mit den Pattern-Kandidaten für Videolernumgebungen in Beziehung gebracht werden. In den meisten dieser Fälle (48), handelte es sich um allgemeine Lösungen, die sich prinzipiell auf Videolernumgebungen übertragen lassen und dazu aber präzisiert werden müssten.

Im Zuge des *Pattern Writings* wurden 45 Patterns formuliert und mehrfach iterativ überarbeitet. 19 Patterns konnten zudem im *Shepherdings* und in *Writers' Workshops* bei Pattern-Konferenzen mit Experten diskutiert werden. Schlussendlich wurden die Patterns in einer Pattern Language miteinander verknüpft. Innerhalb der Mustersprache sind die Patterns zum einen hinsichtlich Mikro- und Makrointeraktivität und zum anderen nach ihren Gestaltungszielen (z.B. Kontribution, Layout) strukturiert. Der Schwerpunkt der Pattern Language liegt erstens auf der Betrachtung und Navigation audiovisueller Medien, zweitens auf dem Bearbeiten und Manipulieren von Informationen und drittens auf der Organisation dieser Informationen innerhalb einer Videolernumgebung. Bedingt durch die angewandte Methode des Pattern Minings enthält die Sprache keine konzeptionelle Modelle für Arten von Videolernumgebungen, wie sie van Welie und Veer [2003] für Webseiten formulierte. Außerdem werden in den Patterns keine Lösungen bezüglich des Stils und der Textgestaltung in Lernumgebungen thematisiert.

Hinsichtlich der Zielsetzung dieses Kapitels und zur Beantwortung der ersten Forschungsfrage wurden 45 Interaction Design Patterns für Videolernumgebungen entwickelt. Abgesehen von technischen Perspektiven im Kontext von Hypervideo, Usability und dem WWW wurden dabei insbesondere kognitive Aspekte in der Verarbeitung von audiovisuellen Medien und Besonder-

heiten des online-basierten Lernens berücksichtigt. Die Patterns sind in einer Wissensbasis aufbereitet und lassen sich durch eine Such- und Filterfunktion sowie eine Entscheidungshilfe anwenderfreundlich nutzen<sup>6</sup>

Bislang wurden die Patterns vor allem bei der Konzeption und Entwicklung von sechs eigenständigen Videolernumgebungen für verschiedene Auftraggeber, Zielgruppen und Untersuchungszwecke angewendet (siehe übernächstes Kapitel, Abschnitt 6.3, S. 246). Die technische Grundlage dafür bestand aus einem Software Framework für interaktive Videos, in welchem fast alle Patterns durch modulare Komponenten repräsentiert sind. Abgesehen davon eignen sich die Patterns als Checkliste zur Bewertung bestehender oder in Entwicklung befindlicher Lernumgebungen. Einen dritten Anwendungsbereich stellt das Lernen anhand von Patterns dar, das hier am Beispiel einer strukturierten Gruppendiskussion erläutert wurde.

#### Kernpunkte

- Die Inhaltsanalyse von 121 Videolernumgebungen ergab, dass 86 % der Lernumgebungen weniger als 10 der 50 funktionalen *User Interfaces* Merkmale aufweisen.
- Die Pattern Language für Videolernumgebungen setzt sich aus 45 Interaction Design Patterns zusammen, die jeweils der Mikro- und Makrointeraktivität sowie einem Gestaltungsziel (Grundfunktionen, zeitbasierter Informationszugang, Kontribution, Strukturierung, Selbstorganisation und Layout) zugeordnet sind.
- Die Interaction Design Patterns bilden sowohl funktionale Anforderungen für ein generisches Framework zur Realisation interaktiver Videos, als auch ein Instrument zur Analyse bestehender Videolernumgebungen.

<sup>6</sup>Siehe <http://designing-video-interfaces.nise81.com/> (abgerufen am 13.06.2016).





## Kapitel 5

# Videobasierte CSCL-Scripts

Im vorangegangenen Kapitel wurden wiederkehrende Lösungen für die Gestaltung von Videolernumgebungen vorgestellt. Lernumgebungen wurden vor allem aus einer technischen Perspektive betrachtet, in der viele kleine funktionale Aspekte die Interaktion zwischen Anwender und Videolernumgebung und somit auch das Lernen mit Videos unterstützen. Die in den Interaction Design Patterns zum Ausdruck gekommenen Lösungen bilden Grundbestandteile videobasierter Lernumgebungen, sie vermitteln jedoch nur ansatzweise wie der Lernprozess anhand von Videos didaktisch und organisatorisch gestaltet werden kann. Dieses Kapitel beschäftigt sich mit einem Ansatz für Lehrende, der es ihnen ermöglicht das Lernen mit Videos didaktisch zu gestalten und organisatorisch zu strukturieren. Möglich wird dies durch die Sequenzierung von Aktivitäten und durch eine Rollenfestlegung innerhalb von Gruppen sowie durch bestimmte Interaktionen zwischen den Beteiligten [Dillenbourg und Tchounikine 2007]. Mit Hilfe der in diesem Kapitel vorgestellten videobasierten CSCL-Scripts lassen sich individuelle, kollektive und kollaborative Lernaktivitäten innerhalb von Videolernumgebungen definieren und umsetzen.

Die Zielsetzung der folgenden Ausführung besteht darin, videobasierte CSCL-Scripts begrifflich einzugrenzen und ihre strukturellen Bestandteile und Abläufe zu spezifizieren. Aus dieser Spezifikation soll neben einer formalen Repräsentation der Daten, auch eine Repräsentation für das *User Interface* entwickelt werden, welche die Modellierung generischer Scripts innerhalb des CSCL-Systems ermöglicht. Die Ergebnisse dieses Kapitels stellen zusammengefasst Anforderungen, für die im nachfolgenden Kapitel (Abschnitt 6.2, S. 235) thematisierte Realisierung einer CSCL-Umgebung für videobasierte Scripts dar.

Znächst erfolgt eine begriffliche Einordnung und Motivation für den Einsatz von videobasierter CSCL-Scripts in Bezug auf allgemeine Scripts und CSCL-Scripts. Anhand von Beispielen aus der Literatur werden anschließend drei grundlegende Formen sowie Merkmale videobasierter CSCL-Scripts erläutert. Ausgehend von diesen Beispielen und einer Analyse allgemeiner Scripts, folgt im zweiten Teil des Kapitels eine Spezifikation der Script-Komponenten und -Mechanismen. Mit Hilfe der Spezifikation werden im dritten Teil Ansätze gesucht, um die hier behandelten Scripts innerhalb eines CSCL-Systems als Daten und für den Endanwender in einer grafischen Bedienoberfläche zu modellieren.

# Einführung

## 5.1.1 Hintergrund zu Scripts und CSCL-Scripts

### Scripts

Der Begriff *Scripts* ist je Fachdisziplin unterschiedlich belegt. Der Ursprung der hier verwendeten Auffassung eines Scripts geht auf die von Schank und Abelson [1977] betriebene Forschung im Bereich der Künstlichen Intelligenz zurück. Roger C. Schank und Rober P. Abelson haben in ihrem Buch „Scripts, plans, goals and understanding / an inquiry into human knowledge structures“ nach Schemata gesucht, wie Menschen ihr Wissen über angemessene Verhaltensweisen in alltäglichen Situationen organisieren. Die Autoren führen das Beispiel eines Restaurantbesuchs an, der als beliebig oft wiederholbare Situation ohne vorherige Planung erfolgen kann, da sich alle Beteiligten über den Ablauf und ihre Rolle im Klaren sind. Sie definieren ein Script wie folgt:

„A script is a structure that describes appropriate sequences of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an interconnected whole, and what is in one slots affects what can be in an other. Scripts handle stylized every day situations. They are not subject to too much change, nor do they provide the apparatus for handling totally novel situations. Thus, a script is a predetermined, stereotyped sequence of actions that defines a well-known situation. Scripts allow for new inferences to objects within them just as if these objects had been previously mentioned; objects within a script may take 'the' without explicit introduction because the script itself has already implicitly introduced them.“ [Schank und Abelson 1977, S. 41]

Schank und Abelson [1977, S. 61-66] unterschieden ferner instrumentelle von persönlichen und situationsbezogenen Scripts. Ein instrumentelles Script enthält eine Folge von Handlungsanweisungen, die in der gegebenen Reihenfolge exakt ausgeführt werden müssen, um ein bestimmtes Ziel zu erreichen. Diese Form von Script kennt man in Gestalt von Bedienungsanleitungen oder Kochrezepten. In einem situationsbezogenen Script sind mehrere Personen beteiligt, die jeweils eine bestimmte Rolle ausüben und ein gemeinsames Verständnis vom Ablauf des Geschehens einschließlich der sozialen Interaktionen haben. Ein persönliches Script kann darin eingebettet sein, wobei die persönlichen Intentionen eines Beteiligten den übrigen Akteuren nicht bewusst sein müssen.

Im Bereich der Pädagogik und pädagogischen Psychologie begannen O'Donnel und Dansereau [1992] Scripts zum dyadischen Lernen von Texten einzusetzen. Scripts wurden somit nicht mehr deskriptiv zur Beschreibung alltäglicher, sozialer Situationen genutzt, sondern dienten als präskriptive Methode zur Gestaltung neuer, sozialer Situationen vor dem Hintergrund des Lernens. Seit der Veröffentlichung des Aufsatzes von O'Donnel und Dansereau [1992] fand diese Methode viel Beachtung – mehr noch als das Werk von Schank und Abelson [1977] in der Künstlichen Intelligenz.

Die didaktische Zielsetzung bestand dabei in der Vermittlung einer effektiven Vorgehensweise zur Aneignung von Wissen. Die Beteiligten sollte die durch das Script vorgegebene Vorgehens-

weise nach einigen Wiederholungen so weit verinnerlichen, dass sie die Technik auch ohne Script-Steuerung beherrschen. Diese Form eines Scripts wird als *micro-script* bezeichnet. Im Kontext des E-Learnings handelt es sich um Dialogmodelle in einer Lernumgebung, die sich Lernende schrittweise aneignen [Dillenbourg und Hong 2008]. Die Aktivitäten einzelner Lernender werden dabei besonders betont [Dillenbourg und Tchounikine 2007]. Beispiele hierfür bilden strukturierte Online-Diskussionen, bei denen Lernende dazu angeregt werden, in ihren Diskussionsbeiträgen die Abfolge von Hypothese, Argument und Gegenargument einzuhalten. Durch die Vorgabe eines passenden Satzanfangs („prompt“) versuchte z.B. Trahasch [2006] Beiträge der genannten Kategorien in der richtigen Abfolge zu fördern. Eine andere Möglichkeit besteht darin, dass Lernende ihre Beiträge bei der Erstellung selbst klassifizieren und somit ihr Handeln reflektieren. Im Laufe der Zeit sollten die Lernenden Argumentationsmuster verinnerlichen und eigenständige, d.h. ohne der Hilfestellung von *prompts* und einer Klassifikation, strukturierte Diskussionen führen können.

Im Gegensatz zu den *micro-scripts* bezeichnen *macro-scripts* laut Dillenbourg und Hong [2008] pädagogische Modelle oder Methoden, wie etwa die Sequenzierung von Handlungsanweisungen, die eine Gruppe auszuführen hat. Zu den bekannten Vertretern zählen beispielsweise die Methoden *Think Pair Share*, *Jigsaw*, *Peer Assessment* und *Peer Review* (vgl. z.B. Hernández-Leo et al. [2006]). Makro-Scripts beschreiben Lerngelegenheiten durch Angabe von Aktivitäten, Sequenzen, Rollenverteilungen, Gruppenbildung oder die Verteilung von Lernressourcen. Die beteiligten Lernenden werden durch klare Anweisungen, welche Aufgaben innerhalb welcher Gruppe mit welchen Personen innerhalb eines definierten Zeitfensters zu bearbeiten sind, unterstützt und geleitet. Innerhalb einer Script-Phase können die Lernenden sowohl kollaborativ in Gruppen zusammen lernen, als auch im Kollektiv aller beteiligten Lernenden agieren oder aber individuell lernen. Die Anwendung von Makro-Scripts geht, im Gegensatz zu den Mikro-Scripts, nicht mit dem Ziel einher, den Ablauf des Scripts durch mehrmalige Anwendung zu verinnerlichen.

Der wesentliche Unterschied zwischen Mikro- und Makro-Scripts besteht indes in der Granularität der Aufgaben und der dafür benötigten Bearbeitungszeit. Aufgaben auf Mikroebene sind in wenigen Minuten, die auf Makroebene unter Umständen nach mehrstündiger Arbeit vollendet [Dillenbourg und Hong 2008]. Gleiches gilt für die Scripts als Ganzes. Ein Makro-Script kann sich durch mehrere Phasen durchaus über mehrere Monate erstrecken. Im Rahmen dieser Arbeit liegt der Schwerpunkt auf Makro-Scripts. In Anlehnung an Dillenbourg und Tchounikine [2007] sind deshalb wie folgt definiert:

**Definition 6.** *Scripts strukturieren individuelle und kollaborative Lernprozesse durch eine festgelegte Abfolge von Handlungen, durch gruppenbezogene Rollen und durch eine Beschränkung der Interaktionen zwischen den beteiligten Personen oder Gruppen.*

### CSCL-Scripts

Zur Abgrenzung von Scripts, die in realen Lernräumen, d.h. in der Präsenzlehre funktionieren, werden im Bereich des *Computer Supported Collaborative Learning* gesonderte Scripts für die virtuelle Lehre definiert. Mit Hilfe von Datenstrukturen und Workflows sind diese nicht ohne die Unterstützung eines Computers ausführbar.

**Definition 7.** *CSCL-Scripts bezeichnen die computerbasierte Form eines Scripts.*

Diese spezielle Form schließt eine Anwendung in der Präsenzlehre nicht aus, wobei das Hauptaugenmerk auf dem Einsatz im Blended und E-Learning liegt. Die Computerisierung und die damit verbundene Automatisierung bringt einige Vorteile mit sich. Einerseits sind es Vorteile gegenüber manuellen, d.h. ohne Rechnerunterstützung durchgeführten Scripts, andererseits sind sie den LMS überlegen, die in aufwendiger Weise für das Gruppenlernen konfiguriert werden müssen.

In der Fernlehre oder in standortübergreifenden Lehrveranstaltungen können sich die Teilnehmer nur in Ausnahmen in Präsenz treffen. CSCL-Scripts ermöglichen es, dass die Lernenden trotz der geographischen Distanz gemeinsam in Gruppen lernen und Aufgaben bearbeiten können. Die computerbasierte Form von Scripts erfüllt eine Reihe logistischer Aufgaben, wie das Zeitmanagement (Deadlines, Phasenwechsel), Informationsfluss (z.B. Verteilung oder Aggregation von Lernressourcen und Aufgabenlösungen) [Dillenbourg 2004].

Auch im Hinblick auf die Organisation von Gruppenlernszenarien bieten CSCL-Scripts einige Vorteile. Die Verwaltung von Gruppen und Informationen (Lernressourcen, Aufgaben, Aufgabenergebnisse) wird durch eine zeitlich getaktete Verteilung sowie eine Steuerung des Rechtemanagements erleichtert. Möchte man beispielsweise zu einem bestimmten Zeitpunkt eine Gruppe neu zusammenstellen, aufteilen oder mehrere zusammenlegen, so kann dies im CSCL-Script in der Form definiert werden, dass auch die Sichtbarkeit und die Zugriffsrechte der Lernressourcen und Werkzeuge für die neuen Gruppen geregelt sind. Die zeitliche Taktung schafft zudem Klarheit über die Verteilung von Aufgaben und den für die Bearbeitung zur Verfügung stehenden Zeitfenstern [Dillenbourg 2004]. Ungeachtet der Vorteile einer zeitabhängigen Steuerung, sind auch konditionale Anweisungen im Sinne einer Lernwegsteuerung möglich [Dillenbourg 2002].

Daraus ergibt sich das Potential, Gruppenlernszenarien für eine beliebige Anzahl an Teilnehmern zu organisieren. Für jedes Script existiert jedoch zunächst eine minimale Konfiguration, die wenigstens eine untere Schranke der Teilnehmerzahl vorgibt. Eine Obergrenze hinsichtlich der Teilnehmeranzahl ist aufgrund der Automatisierung der CSCL-Umgebung durch die technische Leistungsfähigkeit (z.B. Arbeitsspeicher und Prozessorleistung des Servers sowie Datenübertragungskapazität) determiniert.

Unabhängig von den technischen Vorkehrungen Gruppenlernszenarien für eine große Teilnehmerschaft zu automatisieren, besteht immer noch die Notwendigkeit die Lernenden angemessen zu betreuen und ihnen Feedback zu ihren Lernleistungen zu geben. Zum Teil lässt sich dieser Betreuungsaufwand durch Peer-Interaktionen reduzieren. Lernende können beispielsweise selbst Lernressourcen entwickeln und Wissen generieren, welches sie in Gruppen miteinander teilen. Auch die Einschätzung von Arbeitsergebnissen inklusive Feedback kann durch Peers

erfolgen. Vielfach hegen die Lernenden jedoch die (berechtigte) Erwartung, eine abschließende Meinung vom Lehrenden als Experten einholen zu können. Im Kontext prüfungsrelevanter Lernleistungen ist eine Bewertung durch Lehrende ohnehin unabdingbar und nicht durch ein computerbasiertes Script zu ersetzen.

Nichtsdestotrotz lassen sich mit Hilfe von CSCL-Scripts sehr komplexe Gruppenlernszenarien verwirklichen. Die Komplexität entsteht z.B. durch die Anzahl an Gruppen je Phase, die Gruppengröße sowie Änderungen der Gruppenkonstellation, aber auch durch erforderliche Interaktionen zwischen den Gruppen. Um einen reibungslosen Ablauf zu gewährleisten sind Kontrollinstrumente für Lehrende sowie Tutoren unabdingbar. Diese umfassen zumindest ein Monitoring laufender Prozesse und eine Darstellung des Arbeitsfortschritts je Gruppe. Durch die zeitliche Strukturierung der Gruppenaktivitäten sind diese für Lernbegleiter leichter nachvollziehbar [Dillenbourg 2004]. In der Regel werden dafür Nutzeraktivitäten bzw. Arbeitsergebnisse aufgezeichnet, aggregiert und aufbereitet dargestellt.

Gruppenlernszenarien können auch für Lernende eine zusätzliche kognitive Belastung darstellen. Es gilt den Ablauf des Szenarios zu verstehen und sich unter Umständen in jeder Phase mit neuen Gruppenmitgliedern veränderten Aufgabenstellungen und Rollenzuweisungen zu stellen. CSCL-Scripts leisten dahingehend zumindest eine Unterstützung, indem sie den Ablauf eines Scripts transparent darlegen und den aktuellen Systemzustand (Phase, Rolle, Gruppenzugehörigkeit) sowie mögliche Interaktionen (z.B. mit Gruppenmitgliedern und Werkzeugen) sichtbar machen.

### 5.1.2 Definition videobasierter CSCL-Scripts

Mit der vorangegangenen Definitionen von Scripts und CSCL-Scripts wurde die Voraussetzung geschaffen, um den Gegenstand der videobasierten CSCL-Scripts genauer eingrenzen zu können. Da es sich dabei offensichtlich um eine medienspezifische Form eines Scripts handelt, stellt sich zunächst die Frage, ob sich andere Scripts auch auf bestimmte Medien beziehen oder für bestimmte Lernformen zugeschnitten sind.

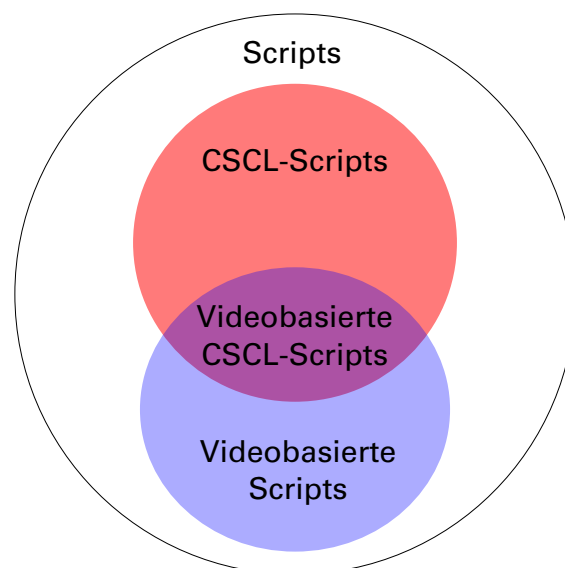
Im überwiegenden Teil der Forschungsarbeiten zu Scripts und CSCL-Scripts finden sich keine Bezüge zum Lernen mit Videos. Hingegen standen vielfach Gruppenlernszenarien im Mittelpunkt, die Lernende dabei unterstützen Texte besser zu verstehen oder Texte gemeinsam zu verfassen (z.B. *Reciprocal Teaching* [Palincsar und Brown 1984], *MURDER* [O'Donnell und Dansereau 1993], *MagicBook* [Dillenbourg 2002], *Social Script* [Weinberger et al. 2005]). Ein Sonderfall stellt die Vermittlung von Lerntexten im Kontext der videovermittelten Kommunikation dar, welche Ertl [2003] mit Hilfe eines *Peer Teaching Scripts* (vgl. O'Donnell und Dansereau [1992]) für Videokonferenzen realisierte.

Wilkinson [1984] und Reynolds [2001] stellen Scripts vor, bei denen Videos als primäre Lernressourcen verwendet werden. Diese Scripts kommen jedoch gänzlich ohne den Einsatz von Computern aus und funktionieren mit beliebigen Wiedergabegeräten wie z.B. einem DVD-Player.

Eine dritte Kategorie bilden kollaborative Lernsettings, die mit Hilfe von Scripts auf die Lösung einer komplexen Problemstellung, wie die Bearbeitung einer Fallstudie oder Programmieraufgabe, abzielen (z.B. *Jigsaw* [Aronson et al. 1978], *UniverSanté* [Berger et al. 2001], *Maze Script* [Miao et al. 2005]).

Abgesehen davon gibt es generischere Scripts, die weder mit einem Medium, noch mit einer Lernform in Zusammenhang stehen (z.B. *ArguGraph* [Jermann und Dillenbourg 1999], *Phase-X* [Engeli 2001]).

Es kann jedoch der Schluss gezogen werden, dass sich viele Scripts und CSCL-Scripts hinsichtlich der eingesetzten Medien bzw. bestimmter Lernformen unterscheiden lassen. Gleichwohl erscheinen die Grenzen als variabel. Es ist möglich Scripts, die für die Präsenzlehre definiert wurden, für den Einsatz im CSCL anzupassen. Beispielsweise geht das Script für Videokonferenzen von Ertl [2003] auf das *MURDER*-Script von O'Donnell und Dansereau [1992] zurück. In ähnlicher Weise lassen sich einige medienspezifische Scripts oder CSCL-Scripts auf das Lernen mit anderen Medien, wie z.B. Videos, übertragen. Die Übertragung des *Jigsaw Puzzles* auf das Lernen mit Videos [Shirouzu und Miyake 2007] ist ein Beispiel dafür. Weitere Beispiele struktureller Ähnlichkeiten werden im Verlauf dieses Kapitels noch erläutert.



**Abbildung 5.1:** Videobasierte CSCL-Scripts als Schnittmenge videobasierter Scripts und CSCL-Scripts

Von diesen bis hierhin noch abstrakten morphologischen Betrachtungen zur Ableitung videobasierter Scripts aus bestehenden Scripts / CSCL-Scripts, lassen sich einfache strukturelle Zusammenhänge erkennen: Videobasierte CSCL-Scripts stellen die Schnittmenge aus CSCL-Scripts und videobasierten Scripts dar (siehe Abb. 5.1). Daraus ergibt sich die folgende Definition.

**Definition 8.** *Videobasierte CSCL-Scripts sind CSCL-Scripts, deren primäre Lernressourcen Videos darstellen.*

Aus dieser einfachen Definition geht noch nicht hervor, welchen Mehrwert videobasierte Scripts haben und worin sie sich, abgesehen von der Einbindung von Videos, von anderen CSCL-Scripts unterscheiden.

### 5.1.3 Mehrwert und Potentiale

Eine Reihe wesentlicher Eigenschaften videobasierter CSCL-Scripts leiten sich aus den übergeordneten CSCL-Scripts ab (siehe Teilabschnitt 5.1.1, S. 174). Aus der Verbindung von Video und CSCL-Script ergeben sich ungeachtet dessen weitere Potentiale, die einen Mehrwert für das Lernen mit Videos darstellen.

Videobasierte CSCL-Scripts zeichnen sich dadurch aus, Videos zu integrieren und selbige den Lernenden nicht als in sich geschlossene, sondern als offene Repräsentation zur Verfügung zu stellen. Informationen im Video werden dadurch für die Lernenden referenzierbar und können Ausgangspunkt für inhaltsbezogene Diskussionen darstellen. Im Kontext des E-Learning stellen Lernvideos somit keine Broadcast-Medien mehr dar, die als kommunikative Einbahnstraße rezeptive Lernformen erzwingen. Desweiteren lassen sich Videos durch verschiedene Formen von Annotationen zu Hypervideos weiterentwickeln, dass heißt auch die Lernenden können sich in die Wissensgenerierung einbringen.

Darüber hinaus ist die Kollaboration sehr eng an die Lernressourcen geknüpft. Die Kollaboration erfolgt anhand der Videos und schließt „co-editing“ und „co-production“ mit ein [Pea und Lindgren 2008]. Das kollaborative Lernen anhand des Videos ist dabei in zweifacher Weise durch ein Script strukturiert.

Erstens wird das Lernen durch das Lernvideo selbst strukturiert. Die inhaltliche Struktur typischer Lernvideos folgt oftmals einem Drehbuch (z.B. Hoppe-Seyler et al. [2014]; Schön und Ebner [2013]; Schwan [2005]). Auch im Fall aufgezeichneter Vortragsreden oder Lehrveranstaltungen unterliegt das Video einer rhetorischen und didaktischen Strukturierung. Während der Rezeption des Videos können die Lernenden zwar wahlfrei navigieren und beispielsweise auf der Zeitleiste hin und her springen, jedoch ist es für das Verständnis der auditiven Informationen unabdingbar, der vorgesehen Abfolge an Informationen zu folgen. Dies ist ein wesentlicher Unterschied zur Rezeption von statischen Medien wie z.B. Text.

Zweitens bauen CSCL-Scripts auf den in sich strukturierten Videos auf und eröffnen interaktive Lernformen. Der Grad der Interaktion kann zwar aufgrund der asynchronen und computervermittelten Kommunikation eingeschränkt sein, jedoch kann mit Hilfe gezielter Instruktionen und der Ermöglichung von Interaktionen mit dem System und zwischen den beteiligten Personen eine virtuelle Zusammenarbeit gelingen. Beispielsweise lassen sich in den Verlauf des Videos Aufrufe einflechten, eine bestimmte Frage zu beantworten bzw. die Antwort eines anderen Gruppenmitglieds zu überarbeiten. Lernende können auch Fragen (oder Überprüfungsfragen) zeitgenau platzieren, auf die andere Lernende dann eingehen. Abgesehen davon lassen sich vorliegende Informationen ergänzen, indem beispielsweise weiterführende Ressourcen verlinkt oder bestimmte Sachverhalte genauer erläutert werden.

Der letztgenannte Aspekt bezeichnet das „co-editing“ von Videos, welche durch Annotationen eine Bearbeitung erfahren. Gleichzeitig kann man auch von einer „co-production“ sprechen, denn das Ergebnis der Annotation ist nicht mehr nur ein Lernvideo, sondern ein gemeinschaftlich erstelltes Hypervideo. Dabei ist zu betonen, dass das eigentliche Video in Gestalt einer Datei in keiner Weise verändert wird. Alle Bearbeitungen werden in der Regel mit Hilfe von Web-



Technologien realisiert. Folglich kann ein Video in mehreren Instanzen auf ganz unterschiedliche Weise durch Annotationen angereichert und auch wiederverwendet werden.

Die eben geschilderte Arbeit mit dem Videomaterial impliziert eine aktive Auseinandersetzung mit den darin vermittelten Lerninhalten. Dies belegen u.a. die vom Autor durchgeführten Feldstudien [Seidel 2013, 2014d].

### 5.1.4 Typisierung videobasierter CSCL-Scripts

Bei der Suche nach beispielhaften Anwendungen videobasierter Scripts finden sich in der Literatur zunächst viele Beispiele, in denen keine kollaborativen Lernformen umgesetzt wurden und das Lernen in Kleingruppen nicht vorgesehen ist. Scripts dieser Art sind sehr einfach gehalten und lassen sich prinzipiell mit Hilfe typischer LMS realisieren. Diese einfachen Scripts verfügen über Möglichkeiten, gewisse Lerneinheiten durch ein Angebot von Lernressourcen (Videos) und Aufgabenstellungen zeitlich zu sequenzieren. Exemplarisch seien an dieser Stelle *xMOOCs* genannt. Den Teilnehmern wird wöchentlich eine bestimmte Anzahl neuer Videolektionen zur Verfügung gestellt. Dazu sollen sie gewisse Übungsaufgaben erledigen. Eine Kollaboration bei der Aufgabenbearbeitung ist dabei nicht ausgeschlossen, sie erfolgt jedoch nicht anhand des Videos, sondern mit Hilfe typischer Werkzeuge der Lernplattform (insb. in Foren). In einigen MOOCs wurden für die Bewertung von Lernleistungen auch Formen des Peer Assessments angewandt [Suen 2014], wobei diese keinen Bezug zu den Videos hatten und folglich nicht als videobasierte Scripts anzusehen sind.

**Tabelle 5.1:** Übersicht von Beispielen für videobasierte CSCL-Scripts. Nebst der Quelle finden sich Angaben zum Modus der Kommunikation (synchron / asynchron) und Einordnung als CSCL-Script.

Quelle des Beispiels	Modus	CSCL	Script-Typ
Lauer und Trahasch [2005a]	asynchron	✓	Videoanalyse und -diskussion
Trahasch [2006]	asynchron	✓	Videoanalyse und -diskussion
Pea und Lindgren [2008]	asynchron	✓	Videoanalyse und -diskussion
Leijen et al. [2009]	asynchron	✓	Videoanalyse und -diskussion
Reinhardt et al. [2011]	asynchron	✓	Videoanalyse und -diskussion
Vohle [2013]	asynchron	✓	Videoanalyse und -diskussion
Seidel [2013]	asynchron	✓	PAPA
Seidel [2014c,d]	synchron und asynchron	✓	PAPA
Wilkinson [1984]	synchron		Video Jigsaw
Reynolds [2001]	synchron		Video Jigsaw
Shirouzu und Miyake [2007]	asynchron	✓	Video Jigsaw

Da sich aus diesen reduzierten Scripts kaum neue Erkenntnisse über die Bestandteile und Mechanismen anspruchsvoller kollaborativer, videobasierter Lernsettings gewinnen lassen, konzentriert sich die folgende Betrachtung auf Settings, in denen die Anwender sehr intensiv mit dem System und den anderen Teilnehmern interagieren. Dazu finden sich relativ wenige Arbeiten von Autoren, die den Aufbau und die Anwendung entsprechender videobasierter Scripts dargelegen. Mit Ausnahme der Untersuchungen von Lauer und Trahasch [2005a] und Trahasch [2006] werden in diesen Arbeiten Scripts nicht explizit genannt. Ferner wurden videobasierte Scripts durch den Autor dieser Arbeit untersucht [Seidel 2013, 2014c,d]. Anhand der in Tab. 5.1 gelisteten elf Beispiele lassen sich strukturelle Ähnlichkeiten der Scripts erkennen, die eine Abstraktion von drei grundlegenden Formen videobasierter CSCL-Scripts ermöglicht. In diesem Teilabschnitt werden diese Script-Typen hinsichtlich ihrer Zielsetzung und ihrem Ablauf genauer erläutert. Eingang finden auch Variationsmöglichkeiten sowie, soweit bekannt, Hindernisse bei der praktischen Umsetzung in der Lehre.

## PAPA Script

*PAPA* steht für *Peer Assessment und Peer Annotation*. Das Script wurde bislang in zwei Feldstudien mit Teilnehmern unterschiedlicher Lehrveranstaltungen, an verschiedenen Hochschulen angewendet und getestet [Seidel 2013, 2014d].

**Ziele** Das didaktische Ziel in der Anwendung des Scripts besteht in erster Linie darin, die Teilnehmer zu einer intensiven Auseinandersetzung mit dem Videomaterial zu bewegen. Bezogen auf ein Video bedeutet das, ein schrittweises Hinzufügen von Aufgaben und den für die Bearbeitung notwendigen Werkzeugen. Die Lernenden analysieren eigenständig den inhaltlichen Aufbau sowie zentrale Begriffe und halten diese fest. In ähnlicher Weise wechseln sie die Rolle vom Lernenden zum Lehrenden und überlegen sich passende Überprüfungsfragen, welche andere Teilnehmer später bearbeiten.

Aus organisatorischer Sicht besteht das Ziel darin, den Zugang zu den in den Videos sequenzierten Informationen durch eine Anreicherung von Kapiteln und Schlagworten zu verbessern. Semantische Informationen dieser Art verbessern die automatische Durchsuchbarkeit eines Videos und erleichtern somit den Umgang mit videografischen Ressourcen. Dies gilt insbesondere für längere Videos mit einer Spieldauer von mehr als 30 Minuten.

Nach Ablauf einer Session können die durch Kapitelmarken, *tags* und Überprüfungsaufgaben angereicherten Lernvideos idealerweise in einem anderen didaktischen Kontext eingesetzt werden.

**Ablauf** Für eine minimale Konfiguration des Scripts (siehe nachfolgende Tab. 5.2) bedarf es vier Teilnehmer und zwei Videos mit einer Spieldauer von mehr als 30 Minuten. Die Teilnehmer teilen sich in zwei Gruppen auf. Jede Gruppe wird in der ersten Script-Phase damit beauftragt, ein Video durch Angabe von Kapitelmarken zu gliedern und mit zeitgenauen Schlüsselworten (*tags*) zu versehen. In der zweiten Phase besteht die Aufgabe darin, drei Überprüfungsfragen (z.B. als *Multiple Choice*, Freitext) im selben Video zu verankern. In Phase 3 erhält jede Gruppe Zugriff auf das von der anderen Gruppe mit Kapiteln, *tags* und Aufgaben annotierte Video. Sie erhalten die Aufgabe, die Überprüfungsfragen im neu hinzugekommenen Video zu bearbeiten.

Währenddessen können sie auch noch weitere Kapitel und *tags* definieren. Anschließend wird in der vierten Phase eine Bilanz gezogen, d.h. die Teilnehmer geben ihren Peers eine Rückmeldung zu den eingereichten Aufgabenlösungen.

**Tabelle 5.2:** Exemplarischer Ablauf eines PAPA-Scripts.

Phase	Gruppe A	Gruppe B	Zeitraum	Kapitelmarken	Tags	Kommentare	Testfrage	Antworten
1. Annotieren	Video 1	Video 2	1.-3. Tag	✓	✓	✓	-	-
2. Testaufgaben festlegen	Video 1	Video 2	4.-5. Tag	✓	✓	✓	✓	-
3. Testaufgaben lösen	Video 2	Video 1	6.-7. Tag	✓	✓	✓	-	✓
4. Lösungsfeedback	bzgl. Gruppe B	bzgl. Gruppe A	7. Tag	✓	✓	✓	-	✓
5. Öffnung	alle Videos	alle Videos	ab 8. Tag	✓	✓	✓	-	✓

**Varianten** In den ersten beiden Phasen können auch mehr als zwei Personen in einer Gruppe zusammengefasst werden, wobei mit zunehmender Gruppengröße die Wahrscheinlichkeit steigt, dass einige Teilnehmer sich nicht mehr an der Annotation des Videos beteiligen können, da ihnen ihre Peers zuvorgekommen sind. Darüber hinaus kann eine Gruppe auch aus einer einzigen Person bestehen, wenngleich sich damit die Kollaboration auf das *Peer Assessment* beschränkt und eine *Peer Annotation* erst ab Phase 3 möglich wird.

Falls mehr als zwei unterschiedliche Videos von mehr als zwei Gruppen gleichzeitig bearbeitet werden, bedarf es einer fünften Script-Phase (Öffnung), in der allen Teilnehmern alle aufbereiteten Videos zur Verfügung stehen. Eine solche Öffnung ist notwendig, um für die Lernenden gleiche Bedingungen zu schaffen und ihnen keine Lernressourcen vorzuenthalten.

In Phase 1 bis 4 können zwei Gruppen durchaus mehr als zwei Videos bearbeiten, wobei die Anzahl bzw. der Umfang oder das Anspruchsniveau der Videos in beiden Gruppen gleich sein sollte.

Die hier dargestellte Form des Peer Assessments erfordert mindestens zwei Videos und zwei Gruppen. Steht jedoch nur ein Video zur Verfügung, kann das Peer Assessment auch innerhalb einer Kleingruppe erfolgen. Jedes Gruppenmitglied kann dabei Aufgaben definieren und dazu eine Musterlösung angeben. Nur der Autor einer Aufgabe kann jedoch deren Inhalt verändern und die richtige Lösung einsehen, während die übrigen Gruppenmitglieder die Fragen beantworten. Folglich gibt es zwei Formen des Peer Assessments: zwischen zwei Gruppen (*inter-group*) und innerhalb einer Gruppe (*intra-group*).

**Analyse** Für die Analyse und Beobachtung des Lernprozesses ist neben einer qualitativen Auswertung der Annotationen inkl. der Aufgabenlösungen auch eine quantitative Evaluation der Videobetrachtung, Gruppeninteraktion und Item-Schwierigkeit der Testfragen möglich. Details dazu finden sich in den zuvor genannten Studien [Seidel 2013, 2014d].

**Hindernisse** Das Peer Assessment verlangt von den Teilnehmern die Fähigkeit, angemessene Überprüfungsfragen zu stellen, die sich sowohl auf den Inhalt des Videos, als auch auf das Vorwissen der Teilnehmer beziehen. Die Erfahrungen mit dieser Form des Assessments haben jedoch gezeigt, dass der Schwierigkeitsgrad der Aufgaben relativ gering ist. Teilnehmer sollten deshalb eine Hilfestellung erfahren, wie sich z.B. mit Hilfe von *Multiple Choice* auch anspruchsvollere Aufgaben und Antwortoptionen definieren lassen, die etwa das Vorwissen einbeziehen.

Abgesehen davon sollten die Teilnehmer darin unterrichtet werden, in welcher Weise sie ihren Peers ein konstruktives und förderliches Feedback zu Aufgabenlösungen geben können.

## Videoanalyse und -diskussion

Die nachfolgende Beschreibung eines Scripts (siehe Tab. 5.3) ist an die von Vohle [2013] vorgestellten Fallbeispiele für Videoanalysen und situationsgenaue Videokommentare in der Musikpädagogik, Landschaftsarchitektur, Lehramtsausbildung, Geographie und Doktorandenausbildung angelehnt. Die Beispiele für Videoanalysen ließen sich nicht exakt hier einfügen, da sie als Blended Learning Szenarien konzipiert sind, wohingegen CSCL-Scripts keine Präsenzelemente beschreiben. Zudem bezieht Vohle die Tutoren viel stärker in den Prozess ein, indem beispielsweise ein Tutor alle Videokommentare erwidert oder die Kommentare mehrerer Teilnehmer zusammenfasst. Eine derart umfangreiche Einbeziehung von Tutoren ist im Hinblick auf skalierbare Teilnehmerzahlen für CSCL-Scripts nicht vorgesehen.

**Ziele** Die didaktische Zielstellung dieser Scripts kann ganz unterschiedlich ausfallen. Je nach Aufgabenstellung besteht der erste Schritt darin das Videomaterial zu Analysieren, um dann die Ergebnisse bzw. Fundstücke im Video zu markieren, um sie im Anschluss reflektieren, in-

**Tabelle 5.3:** Ablauf eines Scripts für Videoanalyse und -diskussion in der minimalen Konfiguration

Phase	Teilnehmer A	Teilnehmer B	Kommentare Markierungen Zusammenfassung
1. Individuelle Analyse / Kategorisierung / Kommentierung	Video 1a	Video 1b	✓ ✓ –
2. Gruppendiskussion der Ergebnisse	Video 1a+1b		✓ ✓ –
3. Zusammenfassung der Ergebnisse	Video 1a+1b		– – ✓

interpretieren oder Zusammenhänge aufzuzeigen zu können. In den genannten Fällen erfolgt jedoch eine Diskussion der Ergebnisse innerhalb einer Gruppe. Die Lernumgebung bietet somit Funktionen, „die den Lernenden aus seiner passiven Rolle des Zuschauers in die eines aktiven Lernenden versetzen, der sich mit seinen Kommilitonen und den Dozierenden über die Inhalte austauschen kann“ [Meyer und Pedrotti 2015].

**Ablauf** In der ersten Phase stellt man den Teilnehmern Videomaterial bereit. Dabei kann es sich um eine Aufzeichnung einer Lehrveranstaltung, eine videografierte Lernleistung eines Lernenden (z.B. videografierte Vorträge / Präsentationen) oder ein sonstiges Filmdokument mit pädagogischem Wert handeln. Die Teilnehmer bekommen die Aufgabe, die für sie relevanten oder hinsichtlich eines Analyseschemas bedeutsamen Stellen im Video farblich zu markieren (z.B. in Form von Ampelfarben oder Pfeilen), zu klassifizieren oder zu kommentieren. Nach dieser individuellen Arbeitsphase finden sich die Teilnehmer in der zweiten Phase in Kleingruppen zusammen. Die Annotationen der einzelnen Teilnehmer werden nun in ein einziges Video integriert, so dass eine gemeinsame Diskussion der Ergebnisse beginnen kann. Durch die Darstellung der farblichen Markierungen auf der Zeitleiste wird die zeitliche Verteilung der als relevant erachteten Szenen sichtbar. Die Teilnehmenden sind nun angehalten, auf die Kommentare ihrer Peers zu antworten. Abschließend fassen sie die Ergebnisse der Phasen 1 und 2 gemeinsam zusammen.

**Varianten** Grundsätzlich sind die Informationen in einem Video nicht nur durch die Angabe eines Zeitpunkts referenzierbar, sondern auch durch einen Zeitraum und darüber hinaus als Fläche, Icon oder Text innerhalb des Videobildes. Falls nur sehr kurze Clips von ein bis zwei Minuten Länge vorliegen, kann die Ressource als Ganzes referenziert werden (z.B. Leijen et al. [2009]).

Abgesehen von der farblichen Markierung einer Szene (z.B. mit Ampelfarben [Vohle 2013]) lassen sich Kommentare und die sich daraus entwickelnden Diskussionen auch auf andere Weisen strukturieren. In den von Trahasch [2006] implementierten Mikro-Scripts werden jeweils neue Diskussionsbeiträge durch Angabe eines Typs („Frage, Antwort, Erklärung, Vorschlag, Klarstellung“) kategorisiert. Je nach ausgewähltem Beitragstyp können Satzanfänge (*prompts*) als Hilfestellung vorgegeben sein [Lauer und Trahasch 2005a].

Eine weitere Variante dieses Scripts wurde mit Hilfe von *DIVER/WebDIVER* umgesetzt und in mehreren Studien untersucht (z.B. Pea und Lindgren [2008]; Zahn et al. [2012]). Dabei werden die Kommentare nicht am Video, sondern an den daraus extrahierten Einzelbildern, Bildausschnitten oder Bewegtbildern annotiert. Somit soll die Aufmerksamkeit für die weiteren Betrachtungen ohne störende Einflüsse durch irrelevante audiovisuelle Informationen erfolgen. Die Kommentare und Antwortkommentare beziehen sich auf das ausgeschnittene Bildmaterial als Ganzes.

Falls durch die Zusammenführung der Kommentare und Markierungen in Phase 2 eine zu große und unübersichtliche Anzahl an Diskussionspunkten entsteht, ist es möglich die Anzahl durch Bildung von Clustern zu reduzieren. Mit Verweis auf den *ArgueGraph* [Jermann und Dillenbourg 1999] kann man auch ausschließlich diejenigen Elemente für die weitere Diskussion

auswählen, bei denen die wenigsten Übereinstimmungen zwischen den Gruppenteilnehmern (z.B. abweichende Kategorisierung oder zeitlicher Abstand) bestehen.

**Analyse** Die Analyse der bei der Script-Durchführung entstehenden Daten betrifft neben dem Nutzerverhalten bei der Videobetrachtung und -annotation vor allem die qualitative Auswertung der Kommentare und Markierungen im Verlauf des Scripts sowie deren Zusammenfassung am Ende. Insbesondere die Verteilung der Annotationen in Abhängigkeit der Medienzeit lässt interessante Vergleiche zwischen einzelnen Lernenden und zwischen unterschiedlichen Gruppen zu.

**Hindernisse** Trotz der hier vorgesehenen Aggregation der individuellen Kommentare und Gegenkommentare bedarf es aus didaktischer Sicht mindestens einer Beurteilung der Arbeitsergebnisse durch einen Lehrenden bzw. Tutor. Ohne eine solche Einschätzung erfahren die Teilnehmer unter Umständen nicht die nötige Anerkennung und inhaltliche Rückmeldungen zu ihren Leistungen.

Wie bereits beim PAPA-Script sollte man die Teilnehmer darin schulen, wie sie ihren Peers ein konstruktives und förderliches Feedback geben können.

## Video Jigsaw

*Jigsaw Puzzles*, im Deutschen auch Gruppenpuzzle genannt, gehen auf Aronson et al. [1978] zurück. Diese Lehrmethode wurden von einer Reihe von Autoren auch für kollaborative Lernszenarien unter Einsatz von Videos adaptiert (z.B. Reynolds [2001]; Wilkinson [1984]) und von Shirouzu und Miyake [2007] zudem in einer virtuellen Lehrveranstaltung erprobt.

**Ziele** Lernende bekommen die Aufgabe eine Problemstellung in Gruppen zu bearbeiten. Die für die Bearbeitung notwendigen Lernressourcen werden unter den Gruppenmitgliedern aufgeteilt, sodass kein Teilnehmer die Aufgabe im Alleingang bewältigen kann. Auf diese Weise wird eine künstliche Informationslücke geschaffen, die nur durch eine Zusammenarbeit geschlossen werden kann. Die Beteiligten sind somit gezwungen, ihr Wissen innerhalb der Gruppe zu teilen und es darüber hinaus den anderen zu vermitteln.

**Ablauf** Mindestens zwei Personen, die sich hinsichtlich eines Merkmals (z.B. kultureller Hintergrund, Geschlecht) unterscheiden, bilden in Phase 1 eine Gruppe. Jedes Gruppenmitglied schaut sich ein anderes Video an. Parallel dazu sollen sich die Lernenden Notizen anfertigen und das Video schriftlich zusammenfassen. Die Inhalte der Videos sind aufeinander abgestimmt und enthalten jeweils essentielle Informationen, ohne die die gegebene Aufgabe oder eine Problemstellung nicht erfolgreich bearbeitet werden kann. Nachdem die Videos angeschaut wurden, formieren sich neue Gruppen für Phase 2. Je Gruppe finden sich die Teilnehmer zusammen, die das gleiche Video gesehen haben. Die Anzahl der Gruppen entspricht somit der Anzahl an unterschiedlichen Videos. In dieser Phase diskutiert jede Gruppe ihre Aufzeichnungen zum betrachteten Video und klärt offene Fragen. Danach formieren sich die Teilnehmer in Phase 3 wieder in der ursprünglichen Gruppenkonstellation und bearbeiten die Problemstellung oder Aufgabe gemeinsam.

**Varianten** In Phase 1 ist es auch gebräuchlich, einen Kurs in so viele Gruppen aufzuteilen, wie Videos zur Verfügung stehen. Auf diese Weise können sie sich unmittelbar austauschen, was eigentlich erst in Phase 2 vorgesehen ist [Reynolds 2001]. Phase 2 ist abgesehen davon zwar ein fester Bestandteil der ursprünglichen Jigsaw-Methode, kann jedoch auch weggelassen werden (z.B. Shirouzu und Miyake [2007]).

Reynolds [2001] konfrontierte im Präsenzunterricht eine Hälfte der Kohorte mit einem Video ohne Ton und die andere Hälfte nur mit dem Ton des Videos. Jeder Teilnehmer sollte sich während der Wiedergabe Notizen machen und das Gehörte bzw. Gesehene im Anschluß innerhalb der Gruppe diskutieren. Nach der Diskussion wurden Zweiergruppen mit je einem Mitglied aus beiden Teilen der Kohorte gebildet. Die Gruppenmitglieder bekamen die Aufgabe, ein gemeinsames Verständnis vom Inhalt des Videos zu entwickeln.

**Analyse** Die Analyse der bei der Script-Durchführung entstehenden Daten betrifft neben dem Nutzerverhalten bei der Videobetrachtung die qualitative Auswertung der zeitgenauen Notizen und Zusammenfassungen in Phase 1 sowie die Aufgabenlösungen in der dritten Phase.

**Hindernisse** Eine Einschränkung erfährt das *Video Jigsaw Script* hinsichtlich der in Phase 3 vorgesehenen Aufgaben, die innerhalb der jeweiligen CSCL-Umgebung auch abgebildet werden müssen. Je nach Aufgabenstellung bedarf es dazu weiterführender Kommunikations- und Arbeitsmittel (z.B. Chat, kollaboratives Schreibwerkzeug oder Spezialsoftware, etwa zur Simulation oder Datenanalyse).

### Spezifikation videobasierter CSCL-Scripts

Dieser Abschnitt widmet sich der Forschungsfrage FF 2a. Die Zielsetzung besteht darin, Komponenten und Mechanismen videobasierter CSCL-Scripts (nachfolgend als Scripts bezeichnet) zu identifizieren und selbige formal zu spezifizieren. Die Spezifikation stellt die Grundlage für die Festlegung von Anforderungen an eine weitgehend generische Videolernumgebung dar, mit deren Hilfe sich Scripts modellieren und ausführen lassen.

Die folgenden Betrachtungen zu videobasierten Scripts gründen auf den Analysen von Kobbe et al. [2007], die aus vier verschiedenen Scripts<sup>7</sup> sechs Script-Komponenten und drei Script-Mechanismen abgeleitet haben. Auch die weiteren Vorarbeiten von Kollar et al. [2006] sowie Piere Dillenbourg und Kollegen (z.B. Dillenbourg [2002]; Dillenbourg und Hong [2008]; Dillenbourg und Tchounikine [2007]) wurden hier berücksichtigt.

Ausgehend von den im vorhergehenden Abschnitt identifizierten Typen videobasierter CSCL-Scripts (vgl. Teilabschnitt 5.1.4, S. 178) und den im Anhang A.16 zusammengefassten CSCL-Scripts (S. 335) werden weitere Lernaktivitäten in Anbetracht der durch die Lernumgebung zur Verfügung gestellten Werkzeuge genauer differenziert. Dabei spielen sowohl die in Abschnitt 4.2.5 (S. 140) beschriebenen Interaction Design Patterns, als auch die von Robert Briggs und Kollegen als *thinkLets* bezeichneten Scripts für *Group Support Systems* eine zentrale Rolle.

---

<sup>7</sup>MURDER Script, Universanté Script, ArgueGraph Script und Social Script.



Im Vergleich zu Kobbe et al. [2007] werden zusätzlich Gruppenformationen als eine Script-Komponente sowie die Verankerung von Aufgaben innerhalb der Lernumgebung bzw. des Videos als ein siebenter Script-Mechanismus thematisiert.

### 5.2.1 Script-Komponenten

Eine Script-Komponente umfasst äquivalent strukturierte Informationseinheiten, die jeweils äquivalente Beziehungen zu Informationseinheiten anderer Komponenten aufweisen. In Bezug auf die hier betrachteten Scripts lassen sich die Komponenten Script, Teilnehmer, Gruppe, Gruppenformation, Lernressource, Software Features und Aufgaben voneinander abgrenzen und in Beziehung setzen.

#### Script

Ein Script  $S$  ist in zeitlich begrenzte und voneinander disjunkte Phasen  $P = \{p_1, p_2, \dots, p_m\}$  unterteilt, wobei  $|P| > 0$ , d.h. mindestens eine Phase definiert ist und die Menge an Phasen je Script endlich ist.

Es gilt  $p_j = (t_{start}, t_{end}) \mid 0 \leq t < \infty$

Diese Phasen folgen einer zeitlichen Ordnung der Art:

$$p_1 < p_2 < \dots < p_m \quad (5.1)$$

In Anlehnung an Dillenbourg [2002] sind einer jeden Phase  $P$  die nachfolgend noch genauer beschriebenen Komponenten Teilnehmer  $T$ , Gruppe  $G$ , Lernressource  $L$ , Feature  $F$  und Aufgabe  $A$  zugeordnet.

Für  $f : P \rightarrow T, G, L, F, A$  gilt:

$$S(p) = T_p, G_p, L_p, F_p, A_p \quad (5.2)$$

.

#### Teilnehmer

Es wird eine endliche Menge aller teilnehmenden Individuen als  $T = \{t_1, t_2, \dots, t_m\}$  definiert. Ein jeder Teilnehmer  $t$  ist durch eine Menge an Kriterien  $t \subseteq K$  charakterisiert mit  $K = \{k_1, k_2, \dots, k_p\} \mid \forall j = 1, \dots, p \text{ mit } k_j \in R^n$ . Die Anzahl an Kriterien und ihre Dimensionalität ist bei jedem Teilnehmer gleich. Burlak [2013] unterscheidet fünf Arten von Kriterien, mit denen sich Teilnehmer charakterisieren lassen:

- fertigkeitsbasierte Kriterien, wie z.B. soziale Beziehungen, Lernleistung;
- eigenschaftsbezogene Kriterien, wie z.B. Geschlecht, kultureller Hintergrund, Sprachkenntnisse, Studiengang, Persönlichkeitsmerkmale;
- prozessbasierten Kriterien, wie z.B. Arbeitszeit im System, kumulierte Wiedergabedauer bzgl. eines oder mehrerer Videos, Anzahl an Beiträgen;
- kontextbezogene Kriterien, wie z.B. Position, Aktivität;
- gruppenbezogene Kriterien, wie z.B. Gruppengröße, Gruppenexistenzzeit.

Jeder Teilnehmer kann zudem eine Rolle  $t \subseteq R, R = \{r_1, r_2, \dots, r_m\}$  wahrnehmen. Diese Rollenzuweisungen können explizit oder implizit erfolgen. Es ist jedoch auch möglich Rollen mit Gruppen gleichzusetzen, indem jeder Gruppe durch Festlegung einer Aufgabe eine spezifische Rollen zugewiesen wird.

### Gruppen und Gruppenformationen

Es wird eine Gruppe  $g$  als eine Menge von Teilnehmern  $t \in T$  definiert, die mindestens zwei Elemente enthält  $|g| > 1$  und in der jeder Teilnehmer  $t_i \in g$  ein Mitglied der Gruppe ist. In manchen Fällen sind auch Gruppen mit nur einem Teilnehmer möglich. Abgesehen von Gruppen unterscheiden Dillenbourg und Hong [2008] die Ebenen („social planes“) des individuellen Lernens und des kollektiven Lernens innerhalb eines gesamten Kurses. Entspricht die Anzahl an Gruppen der Anzahl an Teilnehmern  $|g| = |t|$ , lernen die Teilnehmer individuell. Ist die Anzahl an Gruppen gleich eins  $|g| = 1$  bzw. die Gruppengröße gleich der Teilnehmerzahl, spricht man von einer kollektiven Lernphase.

Eine Gruppenformation  $GF$  bezeichnet eine Menge von Gruppen  $G_{GF} = \{g_1, g_2, \dots, g_m\}$ , für die gilt  $\bigcap G_{GF} = \emptyset$  sowie  $|\bigcup G_{GF}| = |T|$ . Jeder Teilnehmer gehört somit genau einer Gruppe innerhalb der Formation an. Je Script-Phase  $p$  ist immer nur eine Gruppenformation gültig. Eine Formation kann über mehrere oder sogar alle Phasen hinweg bestehen bleiben (z.B. *PAPA-Script* [Seidel 2014d]). Die Scripts *Video-Jigsaw* und *Universanté* sind demgegenüber Beispiele für wechselnde Gruppenkonstellationen bzw. Formationen je Script-Phase.

### Lernressourcen

Es wird eine endliche Menge aller bereitgestellten Lernressourcen als  $L = \{l_1, l_2, \dots, l_m\}$  definiert, wobei  $|L| > 1$  gilt, d.h. mindestens eine Ressource vorhanden sein muss. Im Kontext dieser Arbeit bilden Videos oder Hypervideos die vorrangigen Lernressourcen, wobei nicht nur die vom Lehrenden vorgegebenen Ressourcen dazu zählen, sondern auch der Output der Studierenden (z.B. Annotationen oder Aufgabenlösungen). Eine Lernressource  $l_j \in L$  kann dabei aus einem unteilbaren Element bestehen oder sich aus einer endlichen Menge von Elementen  $S_j = \{s_{j,1}, s_{j,2}, \dots, s_{j,m}\}$  zusammensetzen. Im Folgenden werden diese als Listen bezeichnet. Handelt es sich bei den Elementen einer Liste beispielsweise um Videoannotationen, so stehen diese in Bezug zur Medienzeit in einer zeitlichen Relation (vgl. Allen [1983]) zueinander:

$$s_1 < s_2 < \dots < s_m \quad (5.3)$$

Die folgende Auflistung enthält mögliche Ausprägungen einzelner Elemente, Listen von Elementen sowie verkettete Listen derselben:

- Elemente:
  - Videodatei,
  - Szene eines Videos,
  - Markierung / Hervorhebung im Video (Bildbereich und Zeitabschnitt),
  - Ausschnitt / Extrakt eines zeitlichen und / oder räumlichen Abschnitts im Video,
  - Text (z.B. Schlüsselwort, Kapitelmarke, Kommentar, Aufgabenlösung, Lösungsfeedback),
  - Verweise (z.B. ein benannter Hyperlink),
  - Überprüfungsfrage je Aufgabentyp inklusive Antwortoptionen,
  - Medienelemente (z.B. ein hochgeladenes Bild),
  - sonstiges (z.B. Geo-Koordinaten);
- Listen von Elementen:
  - sortiert (z.B. sortiert nach Entstehungszeitpunkt, Häufigkeit, Relevanz, Korrektheit),
  - unsortiert;
- Verkettete Listen:
  - Klassifikation (z.B. Menge an Kategorien, denen jeweils eine Menge an Elementen zugeordnet ist, etwa Kommentare und Antwortkommentare; Kapitel und Unterkapitel; kategorisierte Schlüsselwörter),
  - Paare,
  - annotiertes Video.

## Software Features

Die Videolernumgebung verfügt über eine Menge an Features  $F = \{f_1, f_2, \dots, f_n\}$ , die durch die Teilnehmer in Anspruch genommen werden können. Für die Modellierung eines Scripts sind nur jene Features relevant, die sich bei Bedarf durch das Script aktivieren und deaktivieren lassen. Die Aktivierung eines Features  $f$  kann sich auf die gesamte Videolernumgebung ( $f \subseteq V$ ) auswirken oder sich auf Rollen ( $f \subseteq R$ ), Lernressourcen ( $f \subseteq L$ ) oder Phasen ( $f \subseteq P$ ) beschränken. Bestimmte Features sind dazu bestimmt Lernressourcen, also sowohl einzelne Elemente als auch Listen, zu erzeugen. Zum Beispiel kann eine Gruppe von Teilnehmern zu einem Video Kapitelmarken annotieren. Aus der Anwendung dieses Features zur Annotation von Kapiteln resultiert am Ende einer Script-Phase ein Inhaltsverzeichnis, also eine sortierte Liste von Kapitelmarken.

Zudem sind Features hinsichtlich der Modi, in denen sie eine Zusammenarbeit ermöglichen zu unterscheiden:

- **asynchron, synchron oder beides:** Können die Teilnehmer unmittelbar, (fast) ohne zeitliche Verzögerung miteinander arbeiten oder ist dies nur zeitversetzt möglich?
- **schriftlich oder mündlich:** Können die Teilnehmer das Feature nutzen, indem sie gemeinsam Texte schreiben / Texte zusammentragen oder können sie auch miteinander sprechen?

## Aufgaben und Aktivitäten

Sei  $\hat{A} = \{\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n\}$  als eine endliche Menge an Aktivitäten und  $O = \{o_1, o_2, \dots, o_m\}$  als eine endliche Menge an Output definiert. Eine Aufgabe  $A = \{a_1, a_2, \dots, a_r\}$  in einer Script-Phase  $p \in P$  ist durch das Tripel von Lernressourcen, Aktivitäten und Output definiert:

$$a_p = (\{l_{p,1}, l_{p,2}, \dots, l_{p,i}\}, \{\hat{a}_{p,1}, \hat{a}_{p,2}, \dots, \hat{a}_{p,j}\}, \{o_{p,1}, o_{p,2}, \dots, o_{p,k}\}) \quad (5.4)$$

Wobei  $i, j, k, p \in \mathbb{N}$ ,  $a_p \in A$ ,  $l_p \in L$ ,  $\hat{a}_p \in \hat{A}$ ,  $o_p \in O$  gilt. Lernressourcen bilden eine Vorbedingung, während der Output als Nachbedingung vom Abschluss der Aktivität abhängt. Eine Aufgabe repräsentiert dadurch einen Aufgabentyp, der durch charakteristische Kombinationen von Lernressourcen, Aktivitäten und Output beschrieben ist. Durch Zusammenlegung von Gruppen oder Zuteilung können Lernressourcen aus den Aktivitäten der vorangegangenen Phase resultieren (vgl. auch Dillenbourg [2002]). Der Output einer Aufgabe bildet somit den Input einer folgenden Aufgabe, d.h. sowohl Input, als auch Output stellen (unter Umständen kleinteilige) Lernressourcen dar.

$$a_{p+1} = ((l_p, \hat{a}_p), \hat{a}_{p+1}, o_{p+1}) \quad (5.5)$$

Dieses Prinzip findet sich u.a. beim *Peer Review*, *Peer Assessment* und bei *Peer Annotation*, aber auch beim *Maze* und *Phase X Script* wieder.

Für eine Spezifikation videobasierter Scripts stellt sich jedoch die Frage, welche Art von Aktivitäten im Kontext von Videolernumgebungen erforderlich sind. Bislang wurden Aktivitäten nur als abstrakte Operatoren betrachtet, die zumindest den Zustand einer Lernressource einschließlich der mit ihr assoziierten Elemente verändern. Nach dieser Festlegung stellt die Betrachtung des Videos keine gesonderte Aktivität dar, da mit ihr keine Zustandsänderungen (z.B. Datenmanipulationen) verbunden sind. Im Kontext des E-Learnings und der Kollaboration im virtuellen Raum lassen sich Zustandsänderungen nur mit Hilfe von Software Features wie z.B. Annotationswerkzeugen bewerkstelligen. Auslöser für die Nutzung solcher Funktionen sind didaktisch motivierte Instruktionen, die vorliegenden Informationen (Input) auf eine bestimmte Weise zu ver- und bearbeiten, sodass ein erwarteter Output entsteht.

Das Zusammenspiel von Features, ihre Konfigurationen und didaktischen Handlungsanweisungen (Script) für die Zusammenarbeit in Gruppen haben Briggs und de Vreede [2009] in so genannten *thinkLets* zusammengefasst. *ThinkLets* dienten ursprünglich dazu, wiederverwendbare und vorhersagbare Gruppeninteraktionen mit Hilfe von *Group Support Software* (z.B. Brainstorming-Werkzeuge) zu ermöglichen. Gegenüber anderen script-artigen Handlungsempfehlungen, wie z.B. den *collaborative learning flow patterns* (CLFP) [Hernández-Leo et al. 2006], bestehen *thinkLets* aus sehr granularen Elementen, aus denen sich komplexere Handlungsanweisungen wie die CLFP zusammensetzen lassen. Für die Kollaboration in Co-Präsenz unterscheiden Briggs und de Vreede [2009] die folgenden Kategorien von *thinkLets*:

- **Divergenz:** Das Ziel dieser Aktivitäten besteht darin, auf Grundlage der vorhandenen Lernressourcen und Output neue Elemente zu generieren. Ein Lernvideo wird somit um weitere Informationen angereichert. Diese Aktivitäten werden in den Patterns für die Annotation von Video thematisiert (vgl. Seidel [2015b]).
- **Konvergenz:** Die Gruppeninteraktion zielt darauf, ein umfangreiches Set an Elementen oder Konzepten für die weitere Arbeit einvernehmlich auf eine relevante Teilmenge zu reduzieren. Die Reduktion auf weniger Elemente mindert die kognitive Last und spart letztendlich Zeit bei der Weiterverwendung der Ergebnisse.
- **Organisation:** Für die weitere Bearbeitung wird eine Menge an Elementen oder Konzepten miteinander in Beziehung gebracht. Dabei gilt es sowohl mögliche Ordnungsstrukturen zu identifizieren, als auch entsprechende Beziehungen herzustellen.
- **Evaluation:** Eine Gruppe kann sich darüber informieren, wie ihre Mitglieder bestimmte Konzepte hinsichtlich gegebener Kriterien einschätzen oder bewerten. Die mit Hilfe von *thinkLets* erzeugten Meinungsbilder helfen bei der Entscheidung eine Auswahl zu treffen.
- **Konsensbildung:** Ist es schwierig innerhalb einer Gruppe einen Konsens zu finden, helfen diese *thinkLets* den Grad der Zustimmung zu erhöhen.

Da die hier betrachteten videobasierten Scripts sich nicht im Bereich der Kollaboration in Co-Präsenz bewegen und somit immanente Bestandteile der *thinkLets*, wie die persönliche Ansprache oder unmittelbare Diskussion, nicht zur Anwendung kommen können, lassen sich nur ausgewählte *thinkLets* für videobasierte Scripts anwenden. Zudem war es erforderlich *thinkLets* für den Anwendungskontext anzupassen oder neu zu kreieren.

Zur Spezifikation von Scripts wurden 23 *thinkLets* entwickelt und adaptiert. Diese finden sich als zusammengefasste Übersicht inklusive Beispiele in Tab. 5.4. Die Aktivitäten sind in die zuvor genannten fünf *thinkLet*-Kategorien unterteilt. Die größte Häufung tritt bei den divergenten Aktivitäten auf, da diese unmittelbar aus den Lösungen für Videoannotationen (Design Patterns) resultieren. Die übrigen Kategorien enthalten nur jene *thinkLets*, die ohne eine geleitete Diskussion in Präsenz funktionieren oder dahingehend angepasst werden konnten.

**Tabelle 5.4:** Übersicht von Aktivitäten unter Angabe der jeweiligen Ein- und Ausgangsgrößen. Die Aktivitäten sind gemäß der Kategorisierung der *thinkLets* gegliedert und enthalten Verweise auf ähnliche *thinkLets* sowie auf passende Interaction Design Patterns für Videolernumgebungen.

<b>Divergente Aktivitäten</b>	
<i>Input:</i>	Video
<i>Aktivität:</i>	Annotation von Kapitelmarken (TABLE OF CONTENT)
<i>Output:</i>	Liste von Elementen, z.B. Kapitelmarken, sortiert nach ihrem zeitlichen Auftreten im Video
<i>Input:</i>	Video   Video mit <i>tags</i>
<i>Aktivität:</i>	Annotation von <i>tags</i> (TEMPORAL TAGS). Beispiele: (1) In einem historischen Filmdokument werden Darsteller und Drehorte im Video markiert und beschriftet, sodass ein Personen- und Ortsregister entsteht. (2) Lernende verschlagworten die in einer Vorlesungsaufzeichnung erläuterten Fachbegriffe. Diese werden zusammengefasst für ein oder mehrere Videos in Form einer <i>tag cloud</i> dargestellt.
<i>Output:</i>	Eine nach Auftrittshäufigkeit oder alphabetisch sortierte Liste von <i>tags</i> .

**Tabelle 5.4:** (Fortsetzung)

<i>Input:</i>	Video   Video mit Kommentaren
<i>Aktivität:</i>	Annotation von Kommentaren (COMMENTS). Beispiele: (1) Lernende platzieren Kommentare und Fragen zeitlich und räumlich im Video. Andere Lernende beziehen sich durch Antwortkommentare darauf, indem sie Kommentare erwidern oder versuchen die Fragen zu beantworten (vgl. auch Lauer und Trahasch [2005a] und Trahasch [2006]). (2) Lernende markieren Bereiche und Zeitabschnitte im Video, um die Aufmerksamkeit auf ausgewählte Aspekte zu lenken und somit eine gemeinsame Grundlage für die von ihnen initiierte Diskussion zu schaffen.
<i>Output:</i>	Eine verkettete Liste von Kommentaren und Antwortkommentaren.
<i>Input:</i>	Video   Video mit Überprüfungsfragen
<i>Aktivität:</i>	Annotation von Überprüfungsfragen (ASSESSMENT). Beispiele: (1) Eine Gruppe von Lernenden annotiert Multiple-Choice-Fragen, die eine andere Gruppe anschließend bearbeitet [Seidel 2013, 2014d]. (2) In einer Gruppe erstellt und verankert jedes Mitglied eine Überprüfungsfrage im Video, welche die übrigen Gruppenmitglieder bearbeiten.
<i>Output:</i>	Liste von Überprüfungsfragen (inkl. Antwortoptionen oder Lösungen), sortiert nach ihrem zeitlichen Auftreten im Video
<i>Input:</i>	Video   Video mit Hyperlinks
<i>Aktivität:</i>	Annotation von Hyperlinks (HYPERLINKS, DETAIL ON DEMAND, BRANCHING VIDEO). Beispiele: (1) Teilnehmer verankern Verweise auf im WWW verfügbare, weiterführende Lernressourcen und Quellen im Video. (2) In einem Vorlesungsarchiv stellen die Lernenden zeitgenaue Verknüpfungen zwischen Aufzeichnungen anderer sowie derselben Lehrveranstaltung her.
<i>Output:</i>	Liste von Hyperlinks (inkl. Verweisziel, Bezeichner, zeitliche Position), sortiert nach ihrem zeitlichen Auftreten im Video
<i>Input:</i>	Video
<i>Aktivität:</i>	Annotation von persönlichen Notizen (USER NOTES). Beispiele: (1) Lernende transkribieren Teile eines Videointerviews, um die Passagen später als Zitate in einer Seminararbeit verwenden zu können. (2) Als Vorbereitung auf die Prüfung fertigen sich Studierende persönliche Aufzeichnung zu den einzelnen Veranstaltungsaufzeichnungen an.
<i>Output:</i>	Liste von Notizen, sortiert nach ihrem zeitlichen Auftreten im Video
<i>Input:</i>	Video
<i>Aktivität:</i>	Annotation von Hervorhebungen im Videobild (VISUAL HIGHLIGHTING). Beispiele: (1) In einem Video wird ein chirurgischer Eingriff vorgeführt. Die Lernenden markieren relevante Vorgänge im Videobild und benennen diese. (2) In einer (nachgestellten) Aufzeichnung eines Team-Meetings markieren die Lernenden Situationen, in denen Konflikte zutage treten und kategorisieren diese anhand typischer Konfliktarten.
<i>Output:</i>	Liste von Hervorhebungen, sortiert nach ihrem zeitlichen Auftreten im Video

**Tabelle 5.4:** (Fortsetzung)

<i>Input:</i>	Video
<i>Aktivität:</i>	Extraktion von räumlichen und zeitlichen Abschnitten (MEDIA FRAGMENTS). Beispiele: (1) In einem 360 °-Video extrahieren Teilnehmer zeitlich begrenzte Bildausschnitte, um die darin zum Ausdruck kommenden Aspekte zur Diskussion zu stellen. (2) Aus einem Multi-Angle Video einer Unterrichtssituation extrahieren Teilnehmer Videofragmente, aus denen die Mitarbeit der im Video sichtbaren Schüler deutlich wird.
<i>Output:</i>	Liste der Extrakte, evtl. sortiert nach ihrem zeitlichen Auftreten im Video
<i>Input:</i>	Video, Liste von Analysekatégorien
<i>Aktivität:</i>	Analyse eines Videos durch zeitliche Verankerung von Kodierungen oder Beschreibungen inhaltlicher, stilistischer oder technischer Aspekte (CLASSIFIED MARKS, MEDIA FRAGMENTS, USER NOTES, COMMENTS). Beispiele: Vohle [2013].
<i>Output:</i>	Zuordnung von Zeitpunkten / Zeitabschnitten im Video und Analysekatégorien
<b>Konvergente Aktivitäten</b>	
<i>Input:</i>	z.B. Liste von Elementen, z.B. Annotationen
<i>Aktivität:</i>	Wähle eine Teilmenge von $n$ Elementen aus einer Liste und verwirfe die übrigen (vgl. <i>thinkLet BroomWagon</i> [Briggs und de Vreede 2009, S. 58]). Beispiel: (1) Aus einer großen Anzahl von im Video verlinkten, weiterführenden Lernressourcen im WWW treffen die Lernenden eine individuelle Auswahl von fünf Links. Die individuellen Bewertungen werden zusammengefasst, um schließlich die fünf am häufigsten ausgewählten Links im Video zu belassen.
<i>Output:</i>	Liste mit Teilmenge an Elementen aus der ursprünglichen Liste von Annotationen
<b>Organisation</b>	
<i>Input:</i>	Liste von Elementen (z.B. Annotationen)
<i>Aktivität:</i>	Identifiziere deckungsgleiche oder identische Items und ersetze diese Paare durch ein einziges Item (vgl. <i>thinkLet Concentration</i> [Briggs und de Vreede 2009, S. 79]) Beispiele: (1) Zwei Gruppen haben ein und dasselbe Video getrennt von einander verschlagwortet und darin weiterführende Links verankert. Nach der Zusammenlegung der beiden Gruppenergebnisse gilt es redundante Annotationen zu entfernen und die Benennung der Link-Label zu vereinheitlichen.
<i>Output:</i>	überarbeitete Liste von Annotationen
<i>Input:</i>	Liste von Elementen (z.B. Annotationen)
<i>Aktivität:</i>	Identifiziere Katégorien oder Hauptthemen (vgl. <i>thinkLet ThemeSeeker</i> [Briggs und de Vreede 2009, S. 84]). Beispiele: (1) In einer aufgezeichneten Vorlesung haben Lernende eine große Anzahl an <i>tags</i> annotiert. Um die Menge an Schlagworten gezielt nutzen zu können, identifizieren sie Oberkatégorien wie Personen, Begriffe, Methoden und geografische Namen. (2) Nach einer lang andauernden Diskussion zu einzelnen Abschnitten eines Videos werden die Lernenden angehalten die Hauptthemen innerhalb der Diskussion zu identifizieren.
<i>Output:</i>	Liste an Katégorien



**Tabelle 5.4:** (Fortsetzung)

<i>Input:</i>	Liste von Elementen (z.B. Annotationen)
<i>Aktivität:</i>	Identifiziere Paare von Elementen und benenne ihre Beziehung (vgl. <i>thinkLet RichRelations</i> [Briggs und de Vreede 2009, S. 86]). Beispiele: (1) Lernende haben eine videografierte Lehrprobe einer Kommilitonin analysiert und individuelle Anmerkungen dazu im Video verankert. Übereinstimmungen in den Einschätzungen mehrerer Lernender werden nun in Gruppenarbeit durch Beziehungen identifiziert und beschrieben.
<i>Output:</i>	Liste an Paaren sowie eine Liste an Kategorien, die sich aus den benannten Beziehungen ergeben
<i>Input:</i>	Liste von Elementen (z.B. Annotationen) und Liste von Kategorien
<i>Aktivität:</i>	Jedes Element einer Liste von $n$ Elementen wird einer von $m$ gegebenen Kategorien zugeordnet (vgl. <i>thinkLet PopcornSort</i> [Briggs und de Vreede 2009, S. 89]). Beispiele: (1) Lernende sortieren alle gleichzeitig die von ihnen annotierten <i>tags</i> den Kategorien Person, Ort und Sonstiges zu.
<i>Output:</i>	Zuordnung von Annotationen und Kategorien
<b>Evaluation</b>	
<i>Input:</i>	Element (z.B. Annotation)
<i>Aktivität:</i>	Freie Bewertung eines Elements (COMMENTS). Beispiel: Offenes Bewerten oder Kommentieren der Aufgabenergebnisse von anderen Lernenden (vgl. auch Ertl und Mandl [2004], Vohle [2013])
<i>Output:</i>	Darstellung der Bewertungen
<i>Input:</i>	Element (z.B. Annotation)
<i>Aktivität:</i>	Bewertung eines Elements auf einer erläuterten Skala von $x$ bis $y$ (USER RATINGS, vgl. <i>thinkLet StrawPoll</i> [Briggs und de Vreede 2009, S. 100]). Beispiele: (1) Lernende bewerten die von ihren Peers eingereichten Lösungen für eine Problemstellung auf einer Skala von 1 bis 5. (2) Eine im Video verankerte Freitextaufgabe wurde von allen Teilnehmern beantwortet. Jeder Lernende kann für die beste der abgegebenen Lösungen einen Punkt vergeben, sodass die Antwort mit den meisten Punkten unmittelbar nach der Frage angezeigt wird. (3) Lernende bewerten die von ihren Peers hochgeladenen Videos danach, wie gut darin ein bestimmter Sachverhalt erklärt ist.
<i>Output:</i>	Zusammenfassung der Bewertungsergebnisse, etwa in Form einer Rangliste
<i>Input:</i>	Element (z.B. Annotation)
<i>Aktivität:</i>	Bewertung eines Elements hinsichtlich mehrere Kriterien (USER RATINGS, vgl. <i>MultiCriteria</i> [Briggs und de Vreede 2009, S. 108] ). Beispiel: Auf einer Skala von 1 bis 5 bewerten die Lernenden jeweils die Verständlichkeit und den Schwierigkeitsgrad von Multiple-Choice-Aufgaben, welche die Gruppenmitglieder im Video verankert haben.
<i>Output:</i>	Bewertungsmatrix

**Tabelle 5.4:** (Fortsetzung)

<i>Input:</i>	Liste von Elementen (z.B. Annotationen)
<i>Aktivität:</i>	Sortiere Elemente einer Liste hinsichtlich eines Kriteriums (vgl. <i>thinkLet BucketShuffel</i> [Briggs und de Vreede 2009, S. 106]). Beispiel: Lernende sortieren die von ihnen gemeinsam annotierten Überprüfungsfragen hinsichtlich der individuell eingeschätzten Schwierigkeit, sodass die schwierigen Aufgaben oben und leichten unten in der Liste stehen.
<i>Output:</i>	Eine hinsichtlich des Kriteriums sortierte Liste
<i>Input:</i>	Liste von Elementen (z.B. Annotationen)
<i>Aktivität:</i>	Wähle aus einer Liste die relevanten oder guten Elemente aus (vgl. <i>thinkLet GoldMiner / Checkmark</i> [Briggs und de Vreede 2009, S. 62 und 111]). Beispiele: (1) Eine Gruppe von Lernenden überarbeitet ein bestehendes Inhaltsverzeichnis und reduziert es auf die wesentlichen Kapitel. (2) Teilnehmer haben die annotierten <i>tags</i> klassifiziert und überprüfen nun für jede Kategorie die Relevanz der <i>tags</i> .
<i>Output:</i>	Hervorgehobene oder separierte Auswahl an Elementen
<i>Input:</i>	Verkettete Liste von Elementen (z.B. Annotationen) und Kategorien der Elemente
<i>Aktivität:</i>	Wähle die Elemente einer Kategorie aus, die nicht zu den übrigen passen und ordne sie einer anderen Kategorie zu (vgl. <i>thinkLet BucketWalk</i> [Briggs und de Vreede 2009, S. 104]). Beispiel: Teilnehmer haben die thematischen <i>tags</i> klassifiziert und überprüfen nun für jede Kategorie die Relevanz der <i>tags</i> .
<i>Output:</i>	Elemente und Kategorien, die übereinstimmend zueinander zugeordnet wurden
<b>Sonstige</b>	
<i>Input:</i>	Video, Elemente (z.B. Aufgabenlösungen)
<i>Aktivität:</i>	Schriftliches Zusammenfassen (oder Wiederholen) der Inhalte einer Szene / eines Videos (COMMENTS, USER NOTES). Beispiele: vgl. Vohle [2013].
<i>Output:</i>	Zusammenfassung eines Videos, der Annotationen eines Videos oder der Kombination aus Video und Annotationen
<i>Input:</i>	Liste von im Video verankerten Überprüfungsfragen
<i>Aktivität:</i>	Bearbeiten oder Beantworten der im Video verankerten Überprüfungsfragen (ASSESSMENT) Beispiele: vgl. Dillenbourg [2002], Seidel [2013].
<i>Output:</i>	Verkettete Liste der eingereichten Lösungen je Überprüfungsfrage
<i>Input:</i>	Liste von im Video verankerten Aufforderungen
<i>Aktivität:</i>	Aufforderung zu einem definierten Zeitpunkt während der Videowiedergabe, den weiteren Fortgang des Videos vorherzusagen. Die Vorhersage kann durch Auswahl von Optionen oder durch eine schriftliche Eingabe erfolgen. [Seidel 2014f]
<i>Output:</i>	Verkettete Liste der eingereichten Vorhersagen je Aufforderung

## 5.2.2 Script-Mechanismen

CSCL-Scripts automatisieren bestimmte Prozesse bei der Konfiguration und Durchführung kollaborativer Lernsettings, die ansonsten manuell ausgeführt werden müssten. Dadurch ist es möglich, Scripts für eine skalierbare Anzahl an Teilnehmern einzusetzen. Als Script-Mechanismen werden jene Routinen bezeichnet, die in script-basierten Lernumgebungen eine Automatisierung ermöglichen. Die Mechanismen regeln beispielsweise die Bildung von Gruppen aus Teilnehmern, die Abfolge und Übergänge der Script-Phasen sowie die Beziehung weiterer Script-Komponenten untereinander. Um diese Mechanismen später in geeigneter Weise formalisieren und in Lernumgebungen abbilden zu können, bedarf es einer näheren Betrachtung ihrer möglichen Ausprägungen.

### Gruppenbildung

In den meisten Fällen werden Gruppen gemäß der Anzahl an Gruppen (z.B. Aufteilung eines Kurses in vier Gruppen), der Gruppengröße (z.B. Aufteilung in Gruppen mit je vier Personen) oder einer Kombination der beiden gebildet [Kobbe et al. 2007]. Nicht selten ist die Gesamtzahl an Teilnehmern nicht ganzzahlig durch die vorgesehene Anzahl an Gruppen oder die Gruppengröße teilbar. Abweichungen in der Gruppengröße sind oftmals nicht zu vermeiden.

Die Aufteilung der Teilnehmer auf die Gruppen kann sequenziell, zufällig oder anhand festgelegter Kriterien erfolgen. Ein Kriterium  $k \in \mathbb{R}^n$  eines Teilnehmers kann zur Bildung homogener oder heterogener Gruppen dienen. Wird beispielsweise das Geschlecht als ein homogenes Kriterium aufgefasst, sind in den Gruppen jeweils nur Personen gleichen Geschlechts. Dient das Geschlecht der Teilnehmer hingegen als heterogenes Kriterium, entstehen gemischte Gruppen, die Geschlechter entweder zu gleichen Teilen oder mit mindestens einem Vertreter enthalten. Konert et al. [2013] haben hierzu eine algorithmische Lösung entwickelt, um mehrere homogene und heterogene Kriterien bei der Formation von Gruppen berücksichtigen zu können.

In den hier betrachteten Scripts wurden bei der Gruppenbildung, abgesehen von der Gruppengröße, externe Kriterien, wie die Zugehörigkeit zu bestehenden Übungsgruppen, der Studiengang und die Nationalität im Hinblick auf Sprachkenntnisse berücksichtigt (z.B. im *UniverSanté* Script). Es können jedoch auch interne Kriterien herangezogen werden, die sich aus dem Lern- oder Nutzungsverhalten sowie aus zurückliegenden Lernergebnissen ermitteln lassen.

Da sich Gruppen je Script-Phase neu formieren können, sind auch Operatoren zur Aufteilung oder, wie im *PAPA-Script*, zur Zusammenlegung von bestehenden Gruppen notwendig (vgl. Seidel [2014c]).

### Sequenzierung

Sequenzierung bezeichnet die Reihenfolge, in der vorgesehene Lernaktivitäten stattfinden sollen. Einzelne Sequenzen bilden Aufgabenblöcke und sind in der Regel zeitlich begrenzt. Aus diesem Grund werden sie als Phasen  $P = \{p_1, p_2, \dots, p_m\}$  bezeichnet, wobei für die Anzahl an Phasen  $M = |P| > 0$  gilt. Der Übergang von einer Phase zur nächsten muss jedoch nicht zwingend an einem Zeitpunkt  $z \in Z$  mit  $f : Z \rightarrow P, z \mapsto p, \forall z > 0$  oder einem Zeitintervall

(z.B. zwei Tage je Phase) festgemacht werden. Phasenübergänge können zum Beispiel auch an Bedingungen  $C = \{c_1, c_2, \dots, c_m\}$  der Art  $p_n \wedge c_n \Rightarrow p_{n+1}$  geknüpft sein. Bedingungen können beispielsweise darin bestehen, eine bestimmte Anzahl an Beiträgen zu verfassen, eine Anzahl an Zeichen zu schreiben oder alle Aufgaben bearbeitet zu haben. Konditionale Sequenzierungen erfordern dabei eine automatische Prüfung, ob oder inwiefern die Bedingung erfüllt ist.

Kobbe et al. [2007] haben folgende fünf Muster in der Sequenzierung von Script-Phasen identifiziert, die sich auch in den hier bislang betrachteten videobasierten Scripts wiederfinden.

- **Branches:** In Abhängigkeit eines Kriteriums variiert das System die Ausgestaltung der folgenden Phase, z.B. durch einen angepassten Schwierigkeitsgrad, eingebaute Wiederholungen, den Umfang an Ressourcen oder die Anzahl verfügbarer Features. Im *Argue-Graph Script* werden jeweils zwei Teilnehmer mit divergierenden Meinungen in einer Gruppe zusammengefasst, damit sie einen Kompromiss finden.
- **Wiederholungen mit geringfügigen Änderungen:** Im *PAPA-Script* dürfen die Lernenden das Video in der ersten Phase nur betrachten, während in der zweiten Phase die Annotation von Kapitelmarken und *tags* und erst in der dritten Phase die Annotation von Überprüfungsfragen möglich ist.
- **Traversion:** Eine Reihe von Lernressourcen werden nacheinander auf die gleiche Art und Weise bearbeitet. Dadurch können Lernende beispielsweise eine Methode anhand mehrerer Datensets oder Beispiele anwenden.
- **Rotation:** Rotation von Rollen findet sich beispielsweise im *PAPA-Script*, wo Lernende im Peer Assessment zunächst in der Rolle eines Lehrenden Überprüfungsfragen definieren und anschließend wieder in die Rolle eines Lernenden wechseln, der Fragen einer anderen Gruppe beantworten soll.
- **Fading:** Schrittweises bzw. phasenweises Hinzufügen oder Entfernen von Features oder Einschränkungen in der Lernumgebung. Im *PAPA-Script* werden je Phase zusätzliche Lernressourcen und Software Features freigeschaltet, bis in der letzten Phase die Lernumgebung inklusive aller Inhalte in vollem Umfang genutzt werden kann.

Der Übergang zwischen zwei Phasen  $p_i$  und  $p_{i+1}$  kann Auswirkungen auf die der Gruppe zugeordneten Lernressourcen haben. Im *PAPA-Script* tauschen zwei Gruppen die zuvor bearbeiteten Videoinstanzen aus, um die dort annotierten Aufgaben zu bearbeiten. Betrachtet man die Zuordnung von Gruppen  $G$  und Lernressourcen  $L$  als eine Matrix  $Z_p : \{1, \dots, n\} \times \{1, \dots, m\}$ , wobei  $n = |G|$ ,  $m = |L|$ ,  $n = m$ , so zirkulieren die Spalten um einen Schritt nach rechts, d.h. eine Gruppe bekommt die Videos ihres linken Nachbarn zugeteilt:

$$Z_{p_i} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \rightarrow Z_{p_{i+1}} = \begin{pmatrix} a_{1n} & a_{11} & \dots & a_{1(n-1)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{mn} & a_{m1} & \dots & a_{m(n-1)} \end{pmatrix} \quad (5.6)$$

Phasenübergänge können sich auch auf die Gruppenkonstellation auswirken. Bleibt eine Gruppe beispielsweise nicht bestehen, wird sie entweder aufgeteilt, zusammengelegt oder vollkommen neu zusammengestellt. Im Folgenden wird in zwei typischen Fällen auf die Verarbeitung der assoziierten Videoannotationen eingegangen.

**Gruppenzusammenlegung:**  $n$  Gruppen haben in der Phase  $p_i$  das gleiche Video in  $n$  von einander getrennten Instanzen annotiert. In der Phase  $p_{i+1}$  werden die  $n$  Gruppen zusammengelegt. Folgende Optionen bestehen für den Fortbestand der Annotationen:

- a) Die Instanzen des Videos mit ihren jeweiligen Annotationen existieren nebeneinander.
- b) Die Annotationen aus allen Videoinstanzen werden in einer Instanz zusammengefasst (vgl. Scripts von Vohle [2013]).
- c) Nur die Differenz der Annotationen wird in einer Videoinstanz abgebildet (vgl. *ArgueGraph* in Jermann und Dillenbourg [1999]).

**Gruppenaufteilung:** Eine Gruppe hat in Phase  $p_i$  ein Video annotiert. Die Gruppe wird in Phase  $p_{i+1}$  in  $m$  Teile aufgeteilt. Folgende Optionen bestehen für den Fortbestand der Videoinstanz:

- a) Die Videoinstanz bleibt für alle zugänglich und editierbar. Alle können an der gleichen Instanz weiterarbeiten.
- b) Die Videoinstanz wird vervielfältigt, so dass jede neue Gruppe unabhängig von den anderen Gruppen mit einer eigenen Kopie weiterarbeiten kann.
- c) Die jeweiligen Gruppenmitglieder bekommen eine Instanz des Videos, in der lediglich ihre persönlichen Beiträge enthalten sind.

### Verknüpfung von Ressourcen und Aktivitäten

Aktivitäten beziehen sich auf bestimmte Lernressourcen  $l \in L$  oder Sektionen  $s \in S$  davon. Um Aufgaben in Verbindung mit videografischen Lernressourcen in Anbetracht ihrer zeitlichen Relation besser beschreiben zu können, hilft es auf die von Allen [1983] postulierte Notation einer intervallbasierten zeitlichen Logik (vgl. S. 67) zurückzugreifen. Mit Hilfe dieser Notation lassen sich nicht nur Relationen im Sinne einer Sequenz von Aufgaben  $a \in A$  und Lernressourcen formulieren, sondern auch Relationen für zeitlich verankerte Aufgaben. Im Kontext videobasierter CSCL-Scripts gibt es drei Relationen:

Mindestens eine Aufgabe folgt auf mindestens eine Lernressource:

$$\{l_{j,1}, l_{j,2}, \dots, l_{j,n}\} < \{a_{j,1}, a_{j,2}, \dots, a_{j,m}\} \mid j, n, m \in \mathbb{N}, |L_j| > 0, |A_j| > 0 \quad (5.7)$$

Mindestens eine Lernressource folgt auf mindestens eine Aufgabe:

$$\{a_{j,1}, a_{j,2}, \dots, a_{j,m}\} < \{l_{j,1}, l_{j,2}, \dots, l_{j,n}\} \mid j, n, m \in \mathbb{N}, |L_j| > 0, |A_j| > 0 \quad (5.8)$$

Mindestens eine Aufgabe ist zeitlich in Sektionen einer Lernressource eingebettet:

$$\{s_{j,1}, s_{j,2}, \dots, s_{j,k}\} < \{a_{j,1}, a_{j,2}, \dots, a_{j,m}\} < \{s_{j,n+1}, s_{j,n+2}, \dots, s_{j,n+k}\} \\ |j, k, n, m \in \mathbb{N}, |L_j| > 0, |A_j| > 0 \quad (5.9)$$

Die zuletzt genannte Relation spielt insbesondere dann eine Rolle, wenn man E-Assessment-Aufgaben im Video verankern möchte. Hinsichtlich weiterführender Konzepte für die zeitliche und räumliche Verankerung verschiedener Aufgabentypen sei an dieser Stelle auf einen Beitrag von Seidel und Jödicke [2014] verwiesen.

## Verankerung und Relation von Aufgaben

In jeder Phase  $p \in P$  hat ein Teilnehmer genau eine Aufgabe zu bearbeiten. Dabei stellt sich die Frage, ob die Aufgabe individuell, gemeinschaftlich oder arbeitsteilig innerhalb einer Gruppe oder kollektiv bearbeitet werden soll. In Anlehnung an die von Dillenbourg [1999] differenzierten „social planes“, bestehen folgende Beziehungen zwischen Aufgaben  $a \in A$ , Teilnehmern  $t \in T$ , Gruppen  $g \in G$  und Rollen  $r \in R$ :

Individuelle Ebene,  $\forall g_p |g_p| = 1$  bzw.  $|G_p| = |T|$ :

- Jeder Teilnehmer hat die gleiche Aufgabe eigenständig zu bearbeiten:  
 $\forall t \exists! a, |A_p| = 1$
- Teilnehmer bearbeiten unterschiedliche Aufgaben:  
 $|A_p| \leq |T|$

Gruppenebene,  $\forall g_p |g_p| > 1$ :

- gemeinschaftliche/kollaborative Bearbeitung innerhalb einer Gruppe:  
 $\forall g_p \exists! a, |A_p| = 1, |r_p| = \emptyset$
- arbeitsteilige, rollenbezogene Bearbeitung je Gruppe, d.h. Teilaufgaben werden innerhalb einer Gruppe verteilt („intra-group distribution“):  
 $\forall t \in g_p \exists! r$
- arbeitsteilige Bearbeitung einer Aufgabe durch mehrere Gruppen („inter-group distribution“):  
 $\forall g_p \exists! r$

Kursebene,  $|g_p| = 1$ :

- kollektive Bearbeitung:  
 $|A_p| = 1, |r_p| = \emptyset$

## Modellierung von CSCL-Scripts

Nach der Beschreibung von Komponenten und Mechanismen videobasierter CSCL-Scripts stellt sich nun die praktische Frage, wie sich diese erstens in einem System und zweitens für den Anwender modellieren lassen. Man unterscheidet hier zwischen der Modellierung einer formalen Repräsentation des Scripts in der ausführenden Anwendung und den Möglichkeiten für den Endanwender, Scripts gemäß ihrer eigenen Anforderungen zu modellieren.

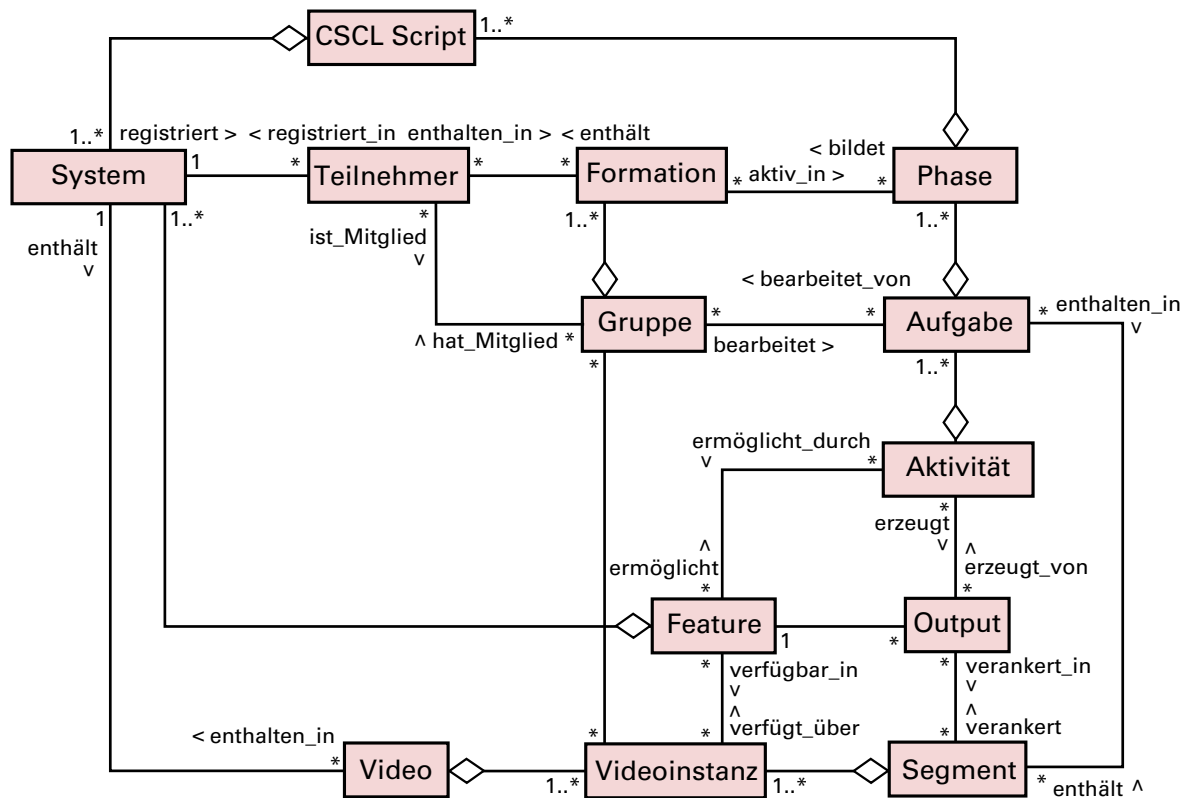
Die Modellierung im System erfordert die Trennung der formalen Repräsentation des Scripts von der Anwendung, die das Script ausführt. Dadurch lassen sich Scripts einfacher austauschen und wiederverwenden [Lauer und Trahasch 2005a]. Welche formale Beschreibung dazu geeignet ist, videobasierte Scripts in Form von Daten und Prozessen zu beschreiben, soll in diesem Teilabschnitt in Bezug auf die Forschungsfrage FF 2b geklärt werden. Dabei sollte ein Modell nach Möglichkeit folgenden Anforderungen genügen:

- **Vollständigkeit:** Das Modell sollte alle Komponenten und Mechanismen videobasierter Scripts enthalten.
- **Skalierbarkeit:** Die formale Repräsentation des Modells sollte hinsichtlich der Anzahl an Scripts, Phasen, Teilnehmern, Videos / Lernressourcen, Gruppen und Features skalieren.
- **Konfigurierbarkeit:** Zudem sollte das Modell anpassbar und innerhalb des Systems wiederverwendbar sein.
- **Interoperabilität:** Idealerweise sollte die Repräsentation sich an bestehenden Standards (z.B. IMS LD) orientieren und auch in anderen CSCL-Systemen genutzt werden können.

Die Modellierung von Scripts aus der Perspektive der Anwender ist ein wesentlicher Aspekt für die Vorbereitung entsprechender Lernszenarien. In der Regel zählen Lehrende zu der Personengruppe, die ein Script festlegen, d.h. Aufgaben definieren, Lernressourcen auswählen, Gruppen bilden und den Zeitrahmen der Phasen bestimmen. Für diese Zielgruppe sind Modellierungssprachen oder Zustandsdiagramme mangels Programmier- und Informatikkenntnisse keine Option. Gefragt sind vielmehr grafische Bedienoberflächen und GUI-Konzepte, welche das formale Modell visuell darstellen und den zeitlichen Ablauf verdeutlichen sowie die Konfiguration und Manipulation der zugrundeliegenden Daten und Prozessen respektive Script-Komponenten und -Mechanismen erlaubt. Somit widmet sich dieser Teilabschnitt der Forschungsfrage FF 2c.

Die Ausgangslage, um Scripts weitgehend generisch innerhalb eines CSCL-Systems zu modellieren ist jedoch relativ komplex. Das Ergebnis der Spezifikation in Abschnitt 5.2 lässt sich wie folgt in einem ER-Diagramm (siehe Abb. 5.2) zusammenfassen. In einer Lernumgebung sind demnach mehrere Videos hinterlegt und Teilnehmer registriert. Die Umgebung unterstützt mehrere Scripts, die sich durch mehrere Phasen auszeichnen, welche wiederum mit Aufgaben und Aktivitäten verbunden sind. In Gruppen formierte Teilnehmer führen Aktivitäten mit Hilfe der zur Verfügung stehenden Features der Lernumgebung aus. Dabei operieren Gruppen auf Instanzen der Videos – bestehend aus mehreren Segmenten, mit denen der Output ihrer Aktivitäten zeitab- oder zeitunabhängig verknüpft ist.





**Abbildung 5.2:** ER-Diagramm der Script-Komponenten und ihrer Beziehungen untereinander

Um ein CSCL-Script in einer Lernumgebung zu realisieren, formulierte Wessner [2012] drei grundlegende Systemansätze:

1. **Statische Systeme:** Ein Script wird statisch im Programmcode verankert und ist durch den Lehrenden als Anwender nur geringfügig anpassbar.
2. **Konfigurierbare Systeme:** Einzelne Script-Komponenten, wie die Auswahl an Lernressourcen oder Werkzeugen, oder die Methode zur Gruppenformation, können durch den Anwender bestimmt werden, während zentrale Mechanismen fest vorgegeben sind.
3. **Generische Systeme:** Mit Hilfe formaler Sprachen, wie z.B. *EML*, *IMS LD* oder *S-BPM*, lassen sich generische Scripts definieren und im CSCL-System zur Laufzeit interpretieren. Dadurch lassen sich Scripts einfacher austauschen und innerhalb unterschiedlicher Systeme wiederverwenden [Lauer und Trahasch 2005a].

Da eine statische Lösung der Zielsetzung von einem möglichst generischen Ansatz widerspricht, gilt es im Verlauf dieses Teilabschnitts zwischen den anderen beiden Ansätzen abzuwägen. Dazu werden im Folgenden existierende CSCL-Systeme betrachtet, bevor dann auf einzelne Modellierungsansätze eingegangen wird. Um die einzelnen Ansätze jedoch im Spannungsfeld ihrer Ausdrucksmächtigkeit und der einfachen Bedienbarkeit besser bewerten zu können, empfiehlt sich eine Unterteilung videobasierter Scripts hinsichtlich verschiedener Komplexitätslevel.

### 5.3.1 Komplexitätslevel

Die Komplexität eines Scripts offenbart sich einem Lehrenden spätestens dann, wenn der Ablauf eines Settings den teilnehmenden Lernenden erläutert werden muss. Die Erfahrungen der zurückliegenden Feldstudien zeigen, dass neben ungleich langen Phasen vor allem wechselnde Gruppenkonstellationen und Aufgabenstellungen je Phase ausführlicher erklärt werden müssen. Letzteres haben auch schon Dillenbourg [2002] in Bezug auf das *UniverSanté Script* bestätigt, bei dem sich die Studierenden und Tutoren über die Komplexität beklagten.

Die Komplexität eines Scripts kommt zuallererst jedoch in der Benutzerschnittstelle zum Ausdruck, mit deren Hilfe ein Script definiert werden soll. Festmachen lässt sich das unter anderem an der Anzahl an notwendigen Arbeitsschritten zur Erreichung eines bestimmten Ziels.

**Tabelle 5.5:** Komplexitätslevel videobasierter CSCL-Scripts

Level	Phasen	Gruppen / Konstellation	Aufgabenbearbeitung	Feature-Konfiguration	Beispiele
0	–	–	–	konstant	Videoportale
1	1.. <i>n</i>	1 konstant	individuell	konstant	xMOOCs
2	1.. <i>n</i>	1.. <i>n</i> (konstant)	individuell intra-group	(konstant)	PAPA-Script (E-Tutoren-Qualif.)
3	1.. <i>n</i>	1.. <i>n</i> wechselnd inkl. Aufteilung und Zusammenlegung	individuell intra-group inter-group	je Gruppe / Phase / Lernressource	PAPA-Script (SCM) [Seidel 2013], Video Jigsaw

In Tab. 5.5 sind vier Komplexitätslevel videobasierter CSCL-Scripts dargestellt. Es wurde bewusst darauf verzichtet die Anzahl bzw. Dauer der verwendeten Lernvideos sowie Anzahl an Teilnehmern einzubeziehen, da Scripts die Verarbeitung und Verwaltung von vielen Ressourcen und Teilnehmern im Wesentlichen automatisieren.

Das niedrigste Level dient als Referenz für Lernumgebungen, die keine Scripts ermöglichen. Lernaktivitäten sind auf Level 0 weder durch zeitlich terminierte Phasen oder Gruppenfestlegungen, noch durch Aufgabenstellungen strukturiert. In Abhängigkeit der verfügbaren Features der Lernumgebung sind jedoch spontane oder von außerhalb oktroyierte kollaborative Lernaktivitäten möglich.

Die Minimalform eines Scripts entspricht Level 1, wenn sich die Teilnehmer für die Dauer der Veranstaltung respektive des Lernszenarios in einer Gruppe zusammenfinden. Da es nur eine Gruppe gibt, besteht die Besonderheit der Gruppenmitgliedschaft lediglich in der Zugangsbeziehung zum System bzw. den Lernressourcen. Die Zeitdauer der Veranstaltung ist begrenzt und zudem in mehrere Phase unterteilt, in denen Aufgaben individuell bearbeitet werden sollen. Typische Vertreter dieser Scripts stellen xMOOCs der Anbieter *edX*, *coursera*, *openHPI* oder *iversity* dar (vgl. Anhang A.5, S. 269).

Das zweite Komplexitätslevel schließt neben der individuellen Bearbeitung von Aufgaben je Phase die gemeinsame Bearbeitung innerhalb der jeweiligen Gruppe von Teilnehmern ein. Die Gruppen agieren somit unabhängig voneinander, ohne dass Ressourcen oder Arbeitsergebnisse ausgetauscht werden. In der Regel bleibt die Gruppenkonstellation auch über den Verlauf des Scripts konstant, sodass auch kein indirekter Austausch durch die Teilnehmer erfolgt. Umgesetzt wurde diese Form zum Beispiel in einem Peer Assessment Script im Rahmen der Qualifizierung von E-Tutoren im Wintersemester 2015 an der TU Dresden (vgl. S. 248).

Level 3 erweitert das vorangegangene Level um die Möglichkeit mehrere Gruppen in den Workflow zur Bearbeitung einer Aufgabe einzubinden. Die im jeweiligen Arbeitsschritt notwendigen Werkzeuge und Software Features können sich je Gruppe oder je Lernressource unterscheiden. Im Verlauf des Scripts wechseln die Teilnehmer zwischen verschiedenen Gruppenformationen hin und her. Das *User Interface* kann sich währenddessen ändern, indem Funktionen wie z.B. die Bearbeitung eines Kommentars oder die Betrachtung eines bestimmten Videos nach einem Phasenübergang nicht mehr möglich sind. Auch wenn die anstehenden Veränderungen im System den Teilnehmern kommuniziert werden, widerspricht es dem Prinzip einer konsistenten Benutzerführung mit einem erwartungskonformen Systemverhalten. Es sind somit Vorkehrungen zu treffen, um die Scriptsteuerung dem Anwender weder als Usability- oder Software Fehler erscheinen zu lassen. Eine Möglichkeit besteht darin, die Menge an Features und Videos grundsätzlich nur auszuweiten, anstatt sie dem Nutzer zu entziehen. Vertreter dieser Art von Scripts sind *Video Jigsaw* sowie eine Variante des PAPA-Scripts [Seidel 2013].

### 5.3.2 Verwandte Systeme und Ansätze zur Modellierung von Scripts

Bei der Betrachtung verwandter CSCL-Systeme interessiert zunächst, in wie weit bestehende Systeme das Lernen mit Videos unterstützen. Falls ein System bereits Videos unterstützt, käme es für eine Weiterentwicklung in Betracht. Unabhängig davon ist in Anbetracht der Forschungsfragen FF 2b/c relevant, auf welche Weise die Scripts formal repräsentiert sind und auf welche Art ihre Modellierung erfolgt. Auch hier besteht das Ziel darin, auf vorhandenen Arbeiten aufbauen zu können und etwa Datenmodelle oder Ansätze zur Modellierung weiterentwickeln oder adaptieren zu können.

Die folgenden CSCL-Systeme sind im Stande CSCL-Scripts zu realisieren und wurden daher für eine nähere Betrachtung ausgewählt:

- **Cooperative OpenWorkflow (COW):** COW ist ein Workflow-System, welches Lernpfade in LMS ermöglicht. Kollaborative Lernaktivitäten werden mit Hilfe „workitems“ realisiert, denen Lernressourcen und Nutzerrollen zugeordnet sind. Für alle Aktivitäten lassen sich verschiedene zeitliche Begrenzungen definieren. Abgesehen vom Workflow-System verfügt COW über kein *User Interface* zur Modellierung von Scripts. [Vantroys und Peter 2003]
- **RELOAD LD Editor und Player**<sup>8</sup> Der LD Editor ermöglicht die Bearbeitung der *IMS LD* Spezifikationen der Levels A, B und C. Dabei wird die XML-Spezifikation in einer grafischen Benutzeroberfläche durch Formulare und Ansichten repräsentiert. Die Desktop-

<sup>8</sup>Siehe <http://www.reload.ac.uk/ldplayer.html> und <http://www.reload.ac.uk/ldeditor.html> (jeweils abgerufen am 26.01.2016).

Anwendung erscheint sehr komplex und für Lehrende ohne vertiefte Kenntnisse des *IMS LD* ungeeignet. Der Learning Design Player basiert auf *CopperCore*<sup>9</sup>

- **Learning Activity Management System (LAMS)**<sup>10</sup>: In einem so genannten *digital lesson plan* werden Sequenzen von Aktivitäten und Lernressourcen für kollaborative Lernaktivitäten definiert. Die Aktivitäten sind durch die Werkzeuge (z.B. Forum, Chat, Abstimmung, Lernressource, Test) repräsentiert. Die einzelnen Aktivitäten werden in einer grafischen Bedienoberfläche linear verknüpft.
- **ManyScripts** ist ein webbasiertes Werkzeug zur Erstellung und Ausführung von Scripts. Es ermöglicht die Konfiguration vordefinierter Scripts, namentlich: *Concept Grid*, *Argue Graph* und *Ice*. Für jedes Script sind spezifische Benutzerdialoge und Ansichten für die Konfiguration und Durchführung implementiert. Vgl. [Dillenbourg 2006].
- **Collage 2 / Web Collage** ist der webbasierte Nachfolger der Desktop-Anwendung *Collage* [Hernández-Leo et al. 2006]. Das System ist an ein LMS angebunden, wodurch sich z.B. Teilnehmerlisten direkt aus den Kursen im LMS erstellen lassen. Auf die gleiche Weise lassen sich Lernressourcen und Werkzeuge in einer Script-Phase referenzieren. Zur Definition einer Script-Phase stehen jeweils so genannte *Collaborative Learning Flow Patterns*<sup>11</sup> zur Verfügung. Die Entscheidung für eines der Flow Patterns wird durch Kontextfragen zu einstellungsbezogenen und prozeduralen Lernzielen sowie durch Angaben zur Art des Problems und der Komplexität unterstützt. Die nacheinander ausgewählten Flow Patterns sind entlang eines linearen Pfades angeordnet. Die je Phase aktiven Teilnehmer müssen manuell ausgewählt werden. Zusätzlich lassen sich Betreuer („Teacher“) festlegen, die die Teilnehmer in einer Phase oder in einem Abschnitt einer Phase unterstützen. Das fertige Script lässt sich als *IMS LD* exportieren und in einem auf die Erfordernisse angepassten LMS ausführen. [Hernández-Leo 2012; Villasclaras-Fernández et al. 2013]

Zusammenfassend ist zunächst festzuhalten, dass die genannten CSCL-Umgebungen keine Kollaboration anhand von Videos unterstützen. Dennoch kommen darin Ansätze zur Modellierung von Scripts zur Anwendung, die hinsichtlich ihrer Passgenauigkeit bzw. Erweiterbarkeit zu prüfen sind. Im Folgenden werden diese und einige wenige weitere Ansätze hinsichtlich ihrer Anwendbarkeit für videobasierte CSCL-Scripts diskutiert.

**IMS Learning Design (LD)** Um computerbasierte Lernszenarien in einer formalen Sprache zu beschreiben, galt *IMS LD* in der Forschung lange Zeit als ein vielversprechender Ansatz. Das Interesse hat jedoch seit 2005 stark nachgelassen<sup>12</sup> *LD* geht über das *IMS Simple Sequencing* in *SCORM* hinaus und fußt auf der Metapher eines Theaters, in dem Lehrende und Lernende Rollen (*roles*) einnehmen und Sequenzen von Handlungen (*activities*) gemeinsam innerhalb der Lernumgebung unter Nutzung von Lernobjekten und Diensten ausführen. Darin inbegriffen sind zeitlich begrenzte und aneinander gereihte Phasen (*acts*), in denen die Handlungen vollführt werden. [Jeffery und Currier 2005]

<sup>9</sup>Siehe <http://coppercore.sourceforge.net/> (abgerufen am 26.01.2016).

<sup>10</sup>Siehe <https://www.lamsfoundation.org/> (abgerufen am 26.01.2016).

<sup>11</sup>Namentlich sind das: Brainstorming, Pyramid, Think Pair Share (TPS), Jigsaw, Think Aloud Pair Problem Solving (TAPPS), Peer Review, Simulation.

<sup>12</sup>In der *ACM Digital Library* finden sich seitdem nur sechs Artikel dazu, wobei der letzte im Jahr 2013 erschienen ist. Vgl. [http://dl.acm.org/results.cfm?query=\"IMS LD\"&srt=publicationDate](http://dl.acm.org/results.cfm?query=\) (abgerufen am 29.04.2016).

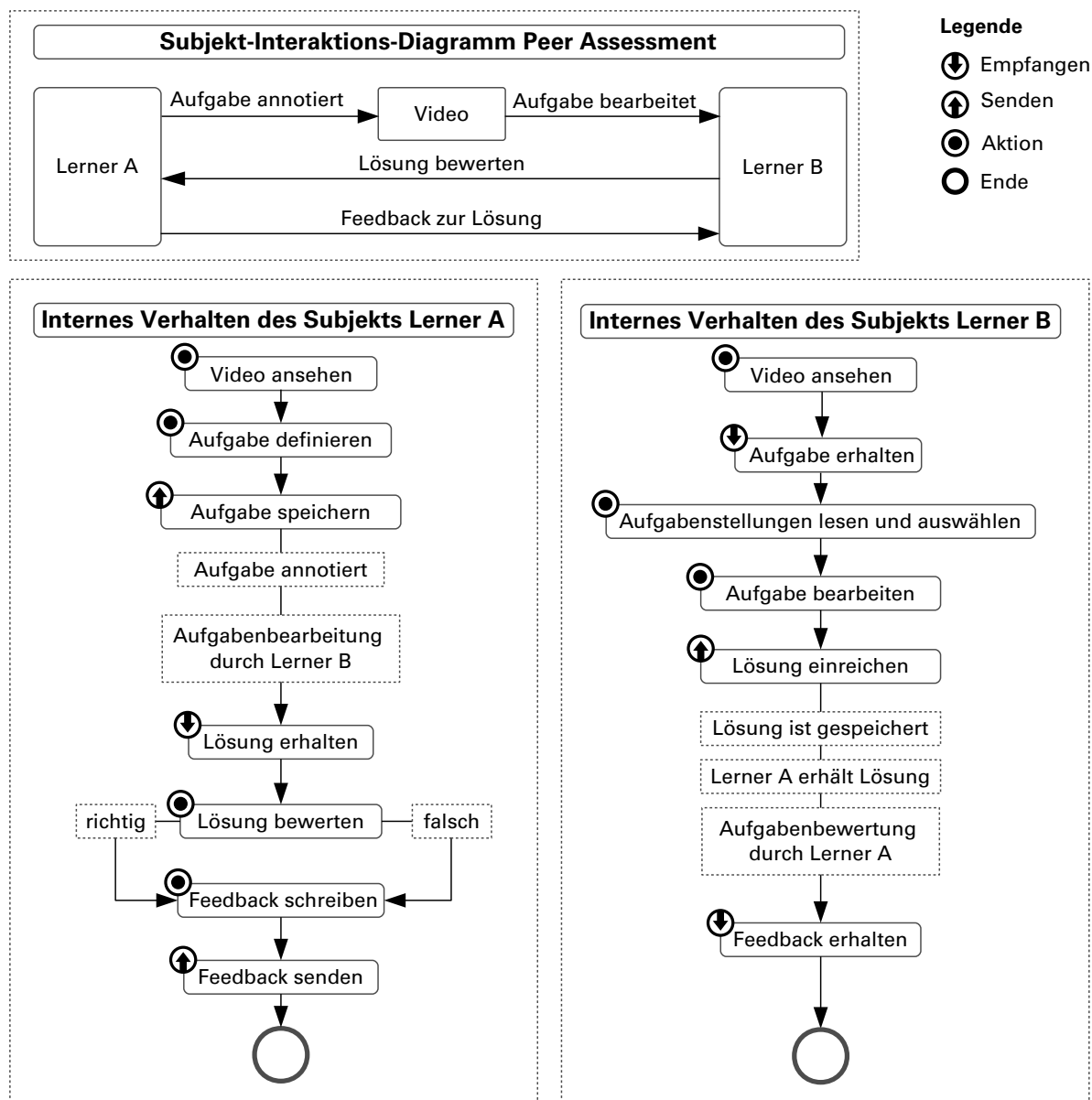
Miao et al. [2005] erachteten LD als eine ungenügende Spezifikation zur formalen Beschreibung von CSCL-Scripts. Durch das Hinzufügen weiterer Entitäten (z.B. *groups*, *artefacts* und *actions*) und das Abrücken von der Theater-Metapher ist zwar eine Integration von Scripts gelungen, doch die ursprüngliche Spezifikation wurde damit überflüssig. Unabhängig davon kritisiert Lonchamp [2006] die überwältigende Komplexität der Spezifikation, in der allein das Metamodell mehr als 40 verschiedene Entitäten und Relationen aufweist, während kollaborative Lernaktivitäten keine Entsprechung finden. Die Notation der Sprache sei überdies zu herausfordernd, um damit Lehrende ohne informatischer Expertise zu behelligen [Lonchamp 2006]. Lösungsansätze bieten die bereits erwähnten grafischen oder formularbasierten Script-Editoren, die zumindest eine Exportfunktion nach LD anbieten (z.B. Harrer et al. [2007, 2009]; Miao et al. [2005]; Villasclaras-Fernández et al. [2013]). Nicht geklärt werden konnte bislang die Frage, inwieweit eine Interoperabilität zwischen den einzelnen Editoren gewährleistet ist.

Folglich erscheint es fraglich, ob *IMS LD* überhaupt noch für die Modellierung von Scripts in Betracht kommen kann. Scripts auf Level 1 bedürfen keiner komplexen Sprache und können auch mittels *SCORM* und *IMS Simple Sequencing* beschrieben werden, während alle komplexeren Scripts für Videolernumgebungen schlechthin nicht darstellbar sind.

**Subject-oriented Business Process Management (S-BPM):** S-BPM beschreibt einen Ansatz zur Modellierung und Ausführung von Geschäftsprozessen mit Hilfe grafischer Symbole, die an die natürliche Sprache angelehnt sind. Die Modellierungswerkzeuge verfolgen einen graphen- oder textbasierten Ansatz, wobei letzterer auf einer für den menschlichen Betrachter nur schwer lesbaren XML-Notation besteht. Ferner wurde auch ein Ansatz für eine natürlich sprachliche Modellierung entwickelt [Höver et al. 2013]. Die grundsätzliche Anwendbarkeit von S-BPM auf CSCL-Scripts wurde zumindest für ausgewählte Scripts durch Höver und Mühlhäuser [2014] belegt.

Mittels S-BPM kann ein Subjekt eine Reihe von Aktivitäten nacheinander ausführen. Als Subjekt kann man sowohl Studierende oder Tutoren, als auch Gruppen auffassen. Im Vergleich zu anderen Ansätzen der Geschäftsprozessmodellierung gibt es nur drei Typen von Aktivitäten: *send*, *receive* und *action*. Dabei können Bedingungen an die Ausführung einer Aktivität geknüpft sein. Falls mehrere Subjekte die gleichen Handlungen vollführen sollen, so lassen sie sich als *Multi Subject* zusammenfassen. Dies ist relevant, um Handlungsmöglichkeiten von Mitgliedern einer Gruppe zu vereinheitlichen oder um identisch konfigurierte Gruppen in einem *Level 2 Script* einfacher zu modellieren. Unterschiedliche Aktionen von Gruppen oder Individuen können nebeneinander existieren und lassen sich miteinander synchronisieren. In nachstehender Abb. 5.3 ist ein vereinfachtes Script für ein videobasiertes Peer Assessment dargestellt. Zwei Lernende haben je eine Rolle inne. Entweder sie annotieren Aufgaben und bewerten die darauf abgegebenen Lösungen oder sie bearbeiten die mit dem Video verknüpften Aufgaben in Erwartung eines Feedbacks.

Mit Hilfe von S-BPM lassen sich der Fluss und die Verarbeitung von Informationen zwischen den beteiligten Objekten auf unterschiedlichen Detailstufen beschreiben. Aufgaben als Tripel von Lernressourcen, Aktivitäten und Output (vgl. S. 188) lassen sich somit gut sequenzieren. Die Wahl eines angemessenen Detailgrads ist dabei entscheidend für die Festlegung von Abläufen, die das System implizit ausführt. Beispielsweise wird im S-BPM-Beispiel eines Peer



**Abbildung 5.3:** Graphennotation von S-BPM für ein vereinfachtes Peer Assessment Script, in dem ein Lernender A Aufgaben in einem Video annotiert, während der Lernende B diese Aufgaben bearbeitet.

Assessment Scripts in Abb. 5.3 nicht geklärt, ob eine Bewertung der Aufgabenlösungen auch automatisiert durch das System erfolgen kann. Dies zu modellieren, hieße unterschiedliche Aufgabentypen als gesonderte Prozesse zu differenzieren. Ein höherer Detailgrad führt zu einem komplexeren Modell, welches mehr Fläche für seine grafische Darstellung beansprucht. Fraglich ist zudem, inwieweit ein Video als ein Subjekt angesehen werden muss, da Lernenden damit interagieren und ihre Kommunikation mittels der im Video hinterlegten Annotationen kontextgebunden erfolgt. Wenn eine Lernende ein Kommentar im Video verankert (*send*) wird dieser Teil des Videos jedoch nicht zwingend Gegenstand einer Interaktionen mit den Peers. Ungeachtet der Prozessmodellierung ist es für die Modellierung von Scripts erforderlich, gewisse Parameter für die Ausführung zu definieren. Je Script-Phase müssen beispielsweise Aufgabenstellungen, zeitliche Rahmen sowie die Zugänglichkeit von Ressourcen und die Optionen der einzelnen Features konfiguriert werden. Innerhalb von S-BPM Diagrammen sind solche



Einstellungen nicht vorgesehen. Auch die Informationen über die Lernressourcen, Gruppenformationen oder Teilnehmerdaten sind hier nicht erfasst. S-BPM bieten relativ viel Gestaltungsfreiheit, wodurch inadäquate, d.h. nicht lauffähige Scripts modelliert werden können. Um dem vorzubeugen, sollte die Prozessmodellierung nur für die interne Repräsentation der Scripts verwendet werden. Die aus der grafischen Notation entstehende Komplexität ist für Anwender ohne Kenntnisse dieser Modellierungstechnik wenig intuitiv. In Anbetracht der Vielzahl der zu definierenden Subjekte und Aktionen erscheint der Aufwand in der Erstellung eines Modells recht hoch.

**Deterministische Automaten** Trahasch [2006] repräsentierte strukturierte Diskussionen mit Hilfe deterministischer Automaten. Durch Angabe von fünf bis sechs Zuständen ließen sich jedoch nur Mikro-Scripts abbilden. Da sich mittels deterministischer Automaten keine nebenläufigen Prozesse abbilden lassen, scheidet dieser Ansatz für komplexere Makro-Scripts der hier vorliegenden Art aus.

**Petri-Netze** Petri-Netze eignen sich zur Abbildung komplexer, nebenläufiger Prozesse. Sie bestehen aus einem Netz aus Zuständen (*places*), in den so genannte Tokens (Marken) platziert werden. Tokens können in einen anderen Zustand übergehen, sofern zwischen beiden eine Verbindung besteht. Mittels so genannter Transitions lassen sich Tokens auch aufteilen und anschließend in zwei Zuständen platzieren. Durch ihre formale Notation sind Petri-Netze gleichzeitig Programmanweisungen für spezielle Ausführungsumgebungen. Prinzipiell sind Petri-Netze ausreichend mächtig, komplexe Scripts abzubilden. Trotz der grafischen Notation erfordern sie vom Anwender ein tiefgreifendes Verständnis der operationalen Anweisungen und Zusammenhänge [Harrer und Malzahn 2006]. Da Laienanwender außerhalb der Informatik damit überfordert sein dürften, ist dieser Ansatz wenig erfolgversprechend [Suthers 2007].

**Grafische Programmierung** Abgesehen von diesen stark formalisierten Ansätzen gibt es Überlegungen, Scripts mit Hilfe grafischer Blöcke zu programmieren (engl. „visual programming“). Programmieren meint hierbei das logische Verknüpfen und Arrangieren von bildlichen Objekten (Blöcke), die jeweils eine oder mehrere ausführbare Operationen repräsentieren. Auf diese Weise lässt sich die komplexe Syntax von Programmier- oder Auszeichnungssprachen vor dem Anwender verbergen. Statt im *IMS LD* oder *S-BPM XML*-Konstrukte erlernen und dann schreiben zu müssen, könnte sich der Entwickler auf die Script-Logik konzentrieren und würde syntaktische Fehler vermeiden. Zudem kann man die Komplexität einer XML-basierten Auszeichnung dadurch reduzieren, indem mehrere Anweisungen zu einem Block zusammengefasst werden. Die Herausforderung besteht jedoch darin, anschauliche und adäquate grafische Objekte zu entwerfen. Darüber hinaus skaliert dieser Modellierungsansatz nur bedingt. Komplexere grafische Konstrukte erfordern viel Darstellungsfläche und sind durch ihre Querverbindungen schwer zu überblicken [Schaefer 2011]. Nichtsdestotrotz bieten Frameworks wie *blockly* [Google 2015] oder *Scratch* (z.B. Peppler und Kafai [2007]) vielversprechende Möglichkeiten, eigene Programmblöcke auch für die Modellierung und Konfiguration von Scripts zu entwickeln.



### 5.3.3 Konzept für eine formale Repräsentation

Die Diskussion im zurückliegenden Teilabschnitt hat gezeigt, dass derzeit weder ein adäquates System, noch eine erprobter Ansatz für die Modellierung videobasierter CSCL-Scripts existiert. Es ist deshalb erforderlich eine individuelle Lösung zu entwickeln, welche die eingangs gestellten Anforderungen hinsichtlich Vollständigkeit, Skalierbarkeit und Konfigurierbarkeit erfüllt. Grundlage für die Festlegung einer formalen Repräsentation sind die Entitäten und Beziehungen im ER-Diagramm in Abb. 5.2 (S. 199). Um das Script von den für die Ausführung relevanten Daten und Inhalten logisch zu trennen, werden die Entitäten in drei Bereiche unterteilt und getrennt behandelt:

- **Teilnehmerverwaltung:** Teilnehmer, Gruppen, Formationen von Gruppen
- **Lernressourcen:** Video, Videoinstanz, Segmente eines Videos und Output
- **Script:** Phasen, Aufgaben, Aktivitäten und Features

Die Teilnehmerverwaltung ist eine notwendige Voraussetzung für die Organisation von Gruppenarbeit. Im System registrierte Teilnehmer können Bestandteil mehrerer Gruppenformationen sein. In jeder Formation gehören sie genau einer Gruppe an. Diese Formationen lassen sich sowohl innerhalb eines oder mehrerer Scripts mehrfach wiederverwenden. Aus diesen beiden Gründen werden die mit den Teilnehmern, Gruppen und Formationen zusammenhängenden Daten unabhängig vom Script behandelt.

Der zweite Bereich umfasst alle Ressourcen (vorrangig Videos), die im Script genutzt oder aus den Aktivitäten der Gruppen hervorgegangen sind. Lernressourcen in Form von Dateien werden dabei ähnlich einem Objekt in der Objektorientierten Programmierung instantiiert. Aus einer Videodatei lassen sich beliebig viele Videoinstanzen erzeugen. Eine Videoinstanz ist jedoch genau einem Script zugeordnet, wohingegen die Videodateien in mehreren Scripts instantiiert und wiederverwendet werden können. Eine Instanz referenziert beispielsweise eine bestimmte Videodatei und enthält alle Aufgabenergebnisse (Outputs), welche ihr eine Gruppe von Teilnehmern zeitgenau je Segment bzw. Zeitstempel oder zeitunabhängig zugeordnet hat. Eine Videoinstanz entspricht somit einem Hypervideo-Dokument, das neben Metadaten sowohl zeitliche, wie auch räumliche Beziehungen zwischen dem Video und den annotierten Informationen enthält (vgl. Teilabschnitt 3.1.3, S. 65). Das Dokument gliedert sich in einen Teil für Metadaten und einen weiteren Teil für die Videoannotationen. Das verwendete Metadaten-Schema auf Basis von *Dublin Core* (vgl. Abb. 5.4) bezieht sich lediglich auf die Videodatei, ohne zeitbezogene oder zeitabhängige Annotationen. Weiterführende Metadaten-Schemata, wie z.B. *videoMD 2.0*, *PBCore 2.1* oder *METS*, die insbesondere für eine Langzeitarchivierung ausgelegt sind, bieten ebenfalls keine Möglichkeiten zeitliche Ereignisse zu referenzieren. Eine Annotation ist mindestens durch das Tripel von Zeitstempel, Dauer und Inhalt definiert.

Der dritte formal zu beschreibende Bereich umfasst das eigentliche Script und seine strukturellen Bestandteile. Ein Script besteht aus einer Folge von Phasen, für die jeweils Aufgaben festzulegen sind. Die Bearbeitung einer Aufgabe ist mit mindestens einer Aktivität verbunden, die mit Hilfe der vom System bereitgestellten Features ausführbar ist. In jeder Phase müssen zudem die Beziehungen zwischen den Gruppen bzw. Gruppenformationen und den Videos bzw. Videoinstanzen definiert werden. Für ein Script auf Level 2 genügt dabei die Angabe einer

Gruppeninformation und eine Liste der für alle Gruppen gesondert zu instantiierenden Videodateien. In einem Level 3 Script werden für jede Kombination aus Gruppe und Videoinstanz die Zugangsberechtigungen und die verfügbaren Features definiert.

Object																																																									
Name	Value																																																								
_id	572a1a6b7c5fdf281e0ae9e0																																																								
video	/static/uploads/14d1023907b3a84dd276c5dd5a5268bf1458077766625.mp4																																																								
updated_at	2016-05-04T15:51:07.220Z																																																								
created_at	2016-04-04T16:13:17.410Z																																																								
metadata	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>title</td><td>B2B-Marketing</td></tr> <tr> <td>subject</td><td>Marketing</td></tr> <tr> <td>description</td><td>In der vorliegenden Videoaufzeichnung wird das Thema ...</td></tr> <tr> <td>type</td><td>video</td></tr> <tr> <td>source</td><td><a href="https://www.youtube.com/watch?v=kQqktXqAqTE">https://www.youtube.com/watch?v=kQqktXqAqTE</a></td></tr> <tr> <td>coverage</td><td>university lecture</td></tr> <tr> <td>creator</td><td>Prof. Dr. Daniel Wentzel</td></tr> <tr> <td>publisher</td><td>RWTH Aachen</td></tr> <tr> <td>contributor</td><td>-</td></tr> <tr> <td>rights</td><td>Creative Commons 2.0</td></tr> <tr> <td>date</td><td>01-01-2014</td></tr> <tr> <td>format</td><td>video/mp4</td></tr> <tr> <td>identifier</td><td>/video/view/572a1a6b7c5fdf281e0ae9e0</td></tr> <tr> <td>Audience</td><td>university students, Master-level</td></tr> <tr> <td>language</td><td>German</td></tr> <tr> <td>length</td><td>3240</td></tr> <tr> <td>thumbnail</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_1.png</td></tr> <tr> <td>1</td><td>still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_2.png</td></tr> </table> </td></tr> <tr> <td>keywords</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>B2B</td></tr> <tr> <td>1</td><td>Marketing</td></tr> </table> </td></tr> </table>	Object		Name	Value	title	B2B-Marketing	subject	Marketing	description	In der vorliegenden Videoaufzeichnung wird das Thema ...	type	video	source	<a href="https://www.youtube.com/watch?v=kQqktXqAqTE">https://www.youtube.com/watch?v=kQqktXqAqTE</a>	coverage	university lecture	creator	Prof. Dr. Daniel Wentzel	publisher	RWTH Aachen	contributor	-	rights	Creative Commons 2.0	date	01-01-2014	format	video/mp4	identifier	/video/view/572a1a6b7c5fdf281e0ae9e0	Audience	university students, Master-level	language	German	length	3240	thumbnail	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_1.png</td></tr> <tr> <td>1</td><td>still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_2.png</td></tr> </table>	Array		Index	Value	0	still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_1.png	1	still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_2.png	keywords	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>B2B</td></tr> <tr> <td>1</td><td>Marketing</td></tr> </table>	Array		Index	Value	0	B2B	1	Marketing
Object																																																									
Name	Value																																																								
title	B2B-Marketing																																																								
subject	Marketing																																																								
description	In der vorliegenden Videoaufzeichnung wird das Thema ...																																																								
type	video																																																								
source	<a href="https://www.youtube.com/watch?v=kQqktXqAqTE">https://www.youtube.com/watch?v=kQqktXqAqTE</a>																																																								
coverage	university lecture																																																								
creator	Prof. Dr. Daniel Wentzel																																																								
publisher	RWTH Aachen																																																								
contributor	-																																																								
rights	Creative Commons 2.0																																																								
date	01-01-2014																																																								
format	video/mp4																																																								
identifier	/video/view/572a1a6b7c5fdf281e0ae9e0																																																								
Audience	university students, Master-level																																																								
language	German																																																								
length	3240																																																								
thumbnail	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_1.png</td></tr> <tr> <td>1</td><td>still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_2.png</td></tr> </table>	Array		Index	Value	0	still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_1.png	1	still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_2.png																																																
Array																																																									
Index	Value																																																								
0	still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_1.png																																																								
1	still_14d1023907b3a84dd276c5dd5a5268bf1458077766625_2.png																																																								
keywords	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>B2B</td></tr> <tr> <td>1</td><td>Marketing</td></tr> </table>	Array		Index	Value	0	B2B	1	Marketing																																																
Array																																																									
Index	Value																																																								
0	B2B																																																								
1	Marketing																																																								
assessment	(empty Array)																																																								
comments	(empty Array)																																																								
hyperlinks	(empty Array)																																																								
toc	(empty Array)																																																								

Abbildung 5.4: Grafische Darstellung des Hypervideo-Dokuments<sup>13</sup>.

### 5.3. MODELLIERUNG VON CSCL-SCRIPTS

Um ein bestehendes Script einfacher wiederverwenden zu können, werden zwei Repräsentationen unterschieden: Instanzen und Templates. Eine *Script Instanz* enthält alle für die

Object																																																																													
Name	Value																																																																												
_id	56f2eb14e77d57f72fccc152																																																																												
title	B2B-Marketing_Instance																																																																												
template	56eec1289805f8b744ffd97d																																																																												
created_at	2016-03-23T19:14:28.963Z																																																																												
updated_at	2016-03-23T19:14:28.965Z																																																																												
status	running																																																																												
current_phase	2																																																																												
results	(empty Array)																																																																												
phases	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>title</td><td>Online Phase B2B Marketing</td></tr> <tr> <td>instruction</td><td>--</td></tr> <tr> <td>supplements</td><td>Willkommen! Folgende Lernmaterialien sollten Sie sich anschauen</td></tr> <tr> <td>seq</td><td>0</td></tr> <tr> <td>_id</td><td>56f2eaede77d57f72fccc151</td></tr> <tr> <td>start</td><td>2016-03-17T16:00:00.000Z</td></tr> <tr> <td>end</td><td>2016-03-24T16:00:00.000Z</td></tr> <tr> <td>groupindex</td><td>0</td></tr> <tr> <td>groupformation</td><td>56f4709e3cd2ef49499230d0</td></tr> <tr> <td>widgets</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table> </td></tr> </table> </td></tr> <tr> <td>video_files</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>56e8804a4669000b77121bed</td></tr> </table> </td></tr> </table> </td></tr> </table>	Array		Index	Value	0	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>title</td><td>Online Phase B2B Marketing</td></tr> <tr> <td>instruction</td><td>--</td></tr> <tr> <td>supplements</td><td>Willkommen! Folgende Lernmaterialien sollten Sie sich anschauen</td></tr> <tr> <td>seq</td><td>0</td></tr> <tr> <td>_id</td><td>56f2eaede77d57f72fccc151</td></tr> <tr> <td>start</td><td>2016-03-17T16:00:00.000Z</td></tr> <tr> <td>end</td><td>2016-03-24T16:00:00.000Z</td></tr> <tr> <td>groupindex</td><td>0</td></tr> <tr> <td>groupformation</td><td>56f4709e3cd2ef49499230d0</td></tr> <tr> <td>widgets</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table> </td></tr> </table> </td></tr> <tr> <td>video_files</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>56e8804a4669000b77121bed</td></tr> </table> </td></tr> </table>	Object		Name	Value	title	Online Phase B2B Marketing	instruction	--	supplements	Willkommen! Folgende Lernmaterialien sollten Sie sich anschauen	seq	0	_id	56f2eaede77d57f72fccc151	start	2016-03-17T16:00:00.000Z	end	2016-03-24T16:00:00.000Z	groupindex	0	groupformation	56f4709e3cd2ef49499230d0	widgets	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table> </td></tr> </table>	Array		Index	Value	0	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table>	Object		Name	Value	label	Comments	name	comments	canBeAnnotated	true	widget_options	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table>	Object		Name	Value	hasTimelineMarker	true	timelineSelector	.v12-timeline-bottom	hasMenu	true	menuSelector	#comments	allowReplies	true	allowEditing	true	allowCreation	true	path	/static/img/user-icons/	video_files	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>56e8804a4669000b77121bed</td></tr> </table>	Array		Index	Value	0	56e8804a4669000b77121bed
Array																																																																													
Index	Value																																																																												
0	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>title</td><td>Online Phase B2B Marketing</td></tr> <tr> <td>instruction</td><td>--</td></tr> <tr> <td>supplements</td><td>Willkommen! Folgende Lernmaterialien sollten Sie sich anschauen</td></tr> <tr> <td>seq</td><td>0</td></tr> <tr> <td>_id</td><td>56f2eaede77d57f72fccc151</td></tr> <tr> <td>start</td><td>2016-03-17T16:00:00.000Z</td></tr> <tr> <td>end</td><td>2016-03-24T16:00:00.000Z</td></tr> <tr> <td>groupindex</td><td>0</td></tr> <tr> <td>groupformation</td><td>56f4709e3cd2ef49499230d0</td></tr> <tr> <td>widgets</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table> </td></tr> </table> </td></tr> <tr> <td>video_files</td><td> <table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>56e8804a4669000b77121bed</td></tr> </table> </td></tr> </table>	Object		Name	Value	title	Online Phase B2B Marketing	instruction	--	supplements	Willkommen! Folgende Lernmaterialien sollten Sie sich anschauen	seq	0	_id	56f2eaede77d57f72fccc151	start	2016-03-17T16:00:00.000Z	end	2016-03-24T16:00:00.000Z	groupindex	0	groupformation	56f4709e3cd2ef49499230d0	widgets	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table> </td></tr> </table>	Array		Index	Value	0	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table>	Object		Name	Value	label	Comments	name	comments	canBeAnnotated	true	widget_options	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table>	Object		Name	Value	hasTimelineMarker	true	timelineSelector	.v12-timeline-bottom	hasMenu	true	menuSelector	#comments	allowReplies	true	allowEditing	true	allowCreation	true	path	/static/img/user-icons/	video_files	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>56e8804a4669000b77121bed</td></tr> </table>	Array		Index	Value	0	56e8804a4669000b77121bed						
Object																																																																													
Name	Value																																																																												
title	Online Phase B2B Marketing																																																																												
instruction	--																																																																												
supplements	Willkommen! Folgende Lernmaterialien sollten Sie sich anschauen																																																																												
seq	0																																																																												
_id	56f2eaede77d57f72fccc151																																																																												
start	2016-03-17T16:00:00.000Z																																																																												
end	2016-03-24T16:00:00.000Z																																																																												
groupindex	0																																																																												
groupformation	56f4709e3cd2ef49499230d0																																																																												
widgets	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table> </td></tr> </table>	Array		Index	Value	0	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table>	Object		Name	Value	label	Comments	name	comments	canBeAnnotated	true	widget_options	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table>	Object		Name	Value	hasTimelineMarker	true	timelineSelector	.v12-timeline-bottom	hasMenu	true	menuSelector	#comments	allowReplies	true	allowEditing	true	allowCreation	true	path	/static/img/user-icons/																																						
Array																																																																													
Index	Value																																																																												
0	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>label</td><td>Comments</td></tr> <tr> <td>name</td><td>comments</td></tr> <tr> <td>canBeAnnotated</td><td>true</td></tr> <tr> <td>widget_options</td><td> <table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table> </td></tr> </table>	Object		Name	Value	label	Comments	name	comments	canBeAnnotated	true	widget_options	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table>	Object		Name	Value	hasTimelineMarker	true	timelineSelector	.v12-timeline-bottom	hasMenu	true	menuSelector	#comments	allowReplies	true	allowEditing	true	allowCreation	true	path	/static/img/user-icons/																																												
Object																																																																													
Name	Value																																																																												
label	Comments																																																																												
name	comments																																																																												
canBeAnnotated	true																																																																												
widget_options	<table> <tr> <th colspan="2">Object</th></tr> <tr> <th>Name</th><th>Value</th></tr> <tr> <td>hasTimelineMarker</td><td>true</td></tr> <tr> <td>timelineSelector</td><td>.v12-timeline-bottom</td></tr> <tr> <td>hasMenu</td><td>true</td></tr> <tr> <td>menuSelector</td><td>#comments</td></tr> <tr> <td>allowReplies</td><td>true</td></tr> <tr> <td>allowEditing</td><td>true</td></tr> <tr> <td>allowCreation</td><td>true</td></tr> <tr> <td>path</td><td>/static/img/user-icons/</td></tr> </table>	Object		Name	Value	hasTimelineMarker	true	timelineSelector	.v12-timeline-bottom	hasMenu	true	menuSelector	#comments	allowReplies	true	allowEditing	true	allowCreation	true	path	/static/img/user-icons/																																																								
Object																																																																													
Name	Value																																																																												
hasTimelineMarker	true																																																																												
timelineSelector	.v12-timeline-bottom																																																																												
hasMenu	true																																																																												
menuSelector	#comments																																																																												
allowReplies	true																																																																												
allowEditing	true																																																																												
allowCreation	true																																																																												
path	/static/img/user-icons/																																																																												
video_files	<table> <tr> <th colspan="2">Array</th></tr> <tr> <th>Index</th><th>Value</th></tr> <tr> <td>0</td><td>56e8804a4669000b77121bed</td></tr> </table>	Array		Index	Value	0	56e8804a4669000b77121bed																																																																						
Array																																																																													
Index	Value																																																																												
0	56e8804a4669000b77121bed																																																																												

**Abbildung 5.5:** Grafische Darstellung des Datenmodells für ein Script-Template<sup>14</sup>, welches aus einer Phase besteht, in der eine referenzierte Gruppenformation Kommentare (*comments*) im Video einsehen, annotieren und beantworten kann.

Ausführung notwendigen Informationen, einschließlich der Zeitdauer der Script-Phasen. Sie beinhaltet zudem Verweise auf die Gruppenformationen und verwendeten Videos. Die Zuordnung von Gruppen und Videoinstanzen wird separat in einem Dokument beschrieben. In Abb. 5.5 ist ein Ausschnitt einer einfachen *Script Instanz* grafisch dargestellt.

Ein *Script Template* abstrahiert die für die Durchführung essentiellen Informationen und beinhaltet nur die Struktur der Lehrveranstaltung, welche aus den Aufgabenstellungen, Verweisen auf die Videos (Dateien) sowie nutzbaren Funktionen (Features) – jeweils pro Phase hervorgehen. Ein Template ist folglich Teil einer *Script Instanz*. Konvertierungen zwischen beiden Repräsentationen sind somit möglich.

Das vorgestellte Modell bildet alle spezifizierten Komponenten ab. Gleiches gilt für die formale Repräsentation derjenigen Daten, die aus der Ausführung eines Script-Mechanismus (z.B. Gruppenbildung oder Sequenzierung) resultieren. Hinsichtlich der Anzahl an Phasen, Teilnehmern, Videos, Gruppenformationen, Gruppen und Features skaliert das Modell. Es lassen sich beliebig viele Komponenten formal abbilden. Lediglich die Interoperabilität ist nur eingeschränkt gegeben. Dies liegt zum einen daran, dass bislang noch keine Standards für Hypervideo-Dokumente oder Videoannotationen existieren (vgl. Teilabschnitt 3.3.4, S. 65 sowie Teilabschnitt 3.1.3, S. 65). Zum anderen genügen etablierte Ansätze wie *IMS LD* oder *S-BPM* nicht der Anforderung, die Komponenten videobasierter Scripts vollständig abzubilden. Da jedoch zum gegenwärtigen Zeitpunkt noch kein System existiert, mit dem vergleichbare Scripts realisierbar sind, ist die Interoperabilität vorerst nicht zwingend erforderlich. Inwieweit eine formale Repräsentation eines Scripts durch den Anwender konfiguriert werden kann, ist Thema des folgenden Teilabschnitts.

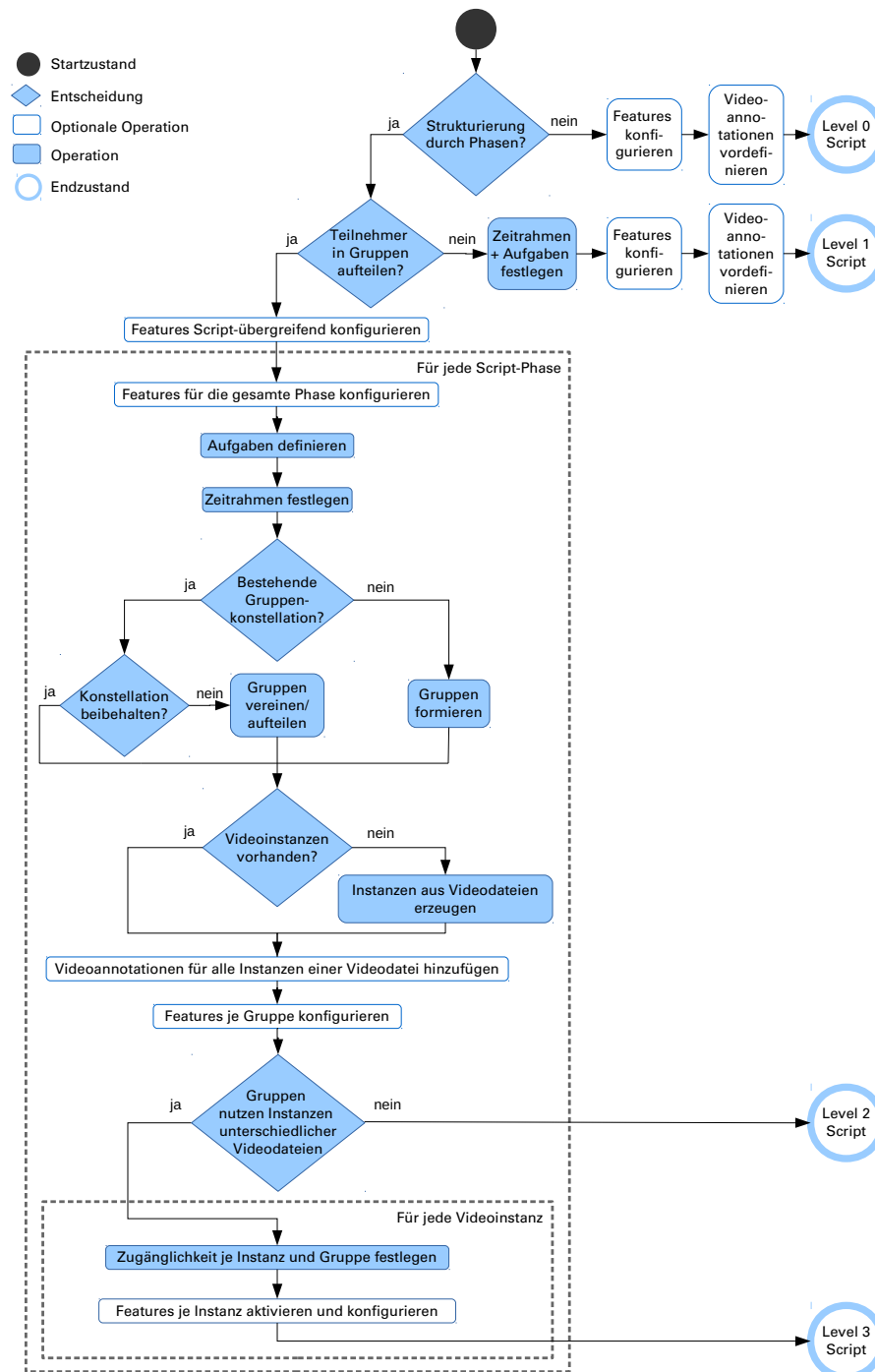
### 5.3.4 Konzept zur Modellierung im User Interface

Nachdem im vorangegangenen Teilabschnitt eine formale Repräsentation videobasierter Scripts vorgestellt wurde, ist nun abschließend zu klären, wie man Endanwender in die Lage versetzt ein Script von Grund auf zu modellieren oder ein vorhandenes gemäß den eigenen Wünschen anzupassen. Ziel soll es sein, dass Anwender alle erforderlichen Daten und Prozesse auch ohne Kenntnis von Auszeichnungs- oder Programmiersprachen in einer grafischen Bedienoberfläche konfigurieren und manipulieren können. Dazu gibt es eine Reihe von Ansätzen, wie zum Beispiel Flussdiagramme in *Collage 2*, Workflow-Ansicht in *COW*, *S-BPM* und *Petri-Netzen* sowie die grafische Programmierung mit *blockly* oder *Scratch*. Mit diesen Konzepten sind jedoch nur die wenigsten Anwender vertraut, da sie von gewohnten Ansätzen der Benutzerführung abweichen.

Miao et al. [2005] hatten deshalb auch einen formularbasierten Ansatz verfolgt. Mit Hilfe einer Baumnavigation ließen sich u.a. Script-Phasen („activities“), Gruppenkonstellationen, Umgebungsvariablen und Ressourcen („artifacts“) in gesonderten Eingabemasken bearbeiten.

Um einen ähnlichen Ansatz auch für die Modellierung von videobasierten CSCL-Scripts nutzen zu können, sind bestimmte Randbedingungen zu berücksichtigen. Entscheidend ist zunächst das Komplexitätslevel. Die Entscheidung darüber kann sich zum einen aus der Auswahl eines Script-Templates ergeben oder während der Erstellung eines Scripts durch Auslassung von Arbeitsschritten getroffen werden. Unabhängig davon sind bestimmte Operationen wie z.B. die

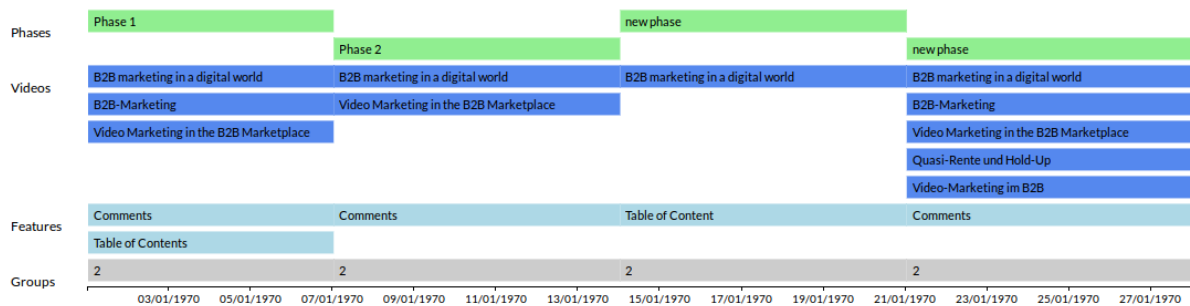
Konfiguration von Features je Video, Gruppe oder Videoinstanz immer optional. Die Existenz mehrerer Phasen und Videoinstanzen bedingt darüber hinaus die Wiederholung sehr ähnlicher Arbeitsschritte. Im Flussdiagramm in Abb. 5.6 wurden diese Randbedingungen bei der Sequenzierung der Arbeitsabläufe berücksichtigt. Für Scripts auf Level 2 oder 3 ist es demnach erforderlich, eine Gruppenformation festzulegen und Videodateien im System zu verankern, aus denen sich u.U. mehrere Videoinstanzen erzeugen lassen.



**Abbildung 5.6:** Entscheidungen bei der Modellierung von Scripts der Komplexitätsstufen 0 – 3.

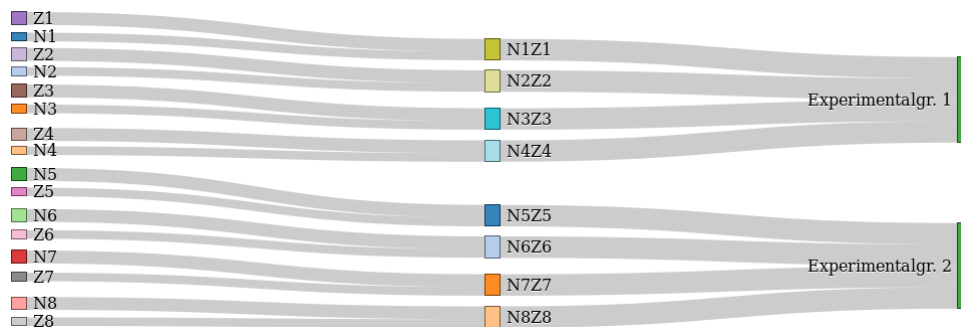
Ergebnisse oder Zwischenergebnisse des Modellierungsprozesses sind dabei nicht unbedingt einfach zu erfassen. Angefangen beim zeitlichen Verlauf und wechselnde Gruppenkonstellationen, bis hin zur Verfügbarkeit von Lernressourcen und Features je Phase können Anwender schnell den Überblick verlieren. Grafische Darstellungen, die ausgewählte Aspekte eines Scripts fokussieren, bieten Anwendern dabei eine Orientierung.

In Abb. 5.7 ist der zeitliche Verlauf der Script-Phasen zu sehen. Auf einem Blick lassen sich wechselnde Gruppenkonstellationen, Lernressourcen und Aufgaben erfassen. Zudem sind zeitliche Überlappungen und Lücken zwischen den Phasen einfacher zu identifizieren.



**Abbildung 5.7:** Visualisierung des zeitlichen Verlaufs der Script-Phasen mit den jeweils verfügbaren Videodateien und Features sowie den damit operierenden Gruppen.

Abb. 5.8 zeigt eine Visualisierung der je Script-Phase wechselnden Gruppenformationen in einer Variante eines Sankey-Diagramms. Im dargestellten Beispiel ist die Tendenz zur Zusammenlegung von Gruppen deutlich zu erkennen. Überdies lassen sich die Gruppenzuordnungen der Teilnehmer im Verlauf des Scripts eindeutig nachvollziehen. Die Dicke der Linien und Knoten repräsentieren dabei die Gruppengröße.



**Abbildung 5.8:** Sankey-Diagramm der über drei Phasen wechselnden Gruppenkonstellationen von Einer- über Zweiergruppen, bis hin zu zwei Gruppen mit jeweils acht Teilnehmern.

Während in Level-2 Scripts alle Gruppen Instanzen der gleichen Videodateien bearbeiten und zudem identisch konfigurierte Features nutzen, können sich die Zugangsberechtigungen und Konfigurationen in Level-3 Scripts je Gruppe und Videoinstanz unterscheiden. Damit Lehrende bei der Erstellung eines Scripts die Übersicht behalten und insbesondere falsche Zuweisungen vermeiden, werden die Gruppen und Videoinstanzen in einer Matrix gegenübergestellt. In den Zeilen der Matrix sind die Gruppen der zuvor ausgewählten Formation dargestellt. In den Spalten lassen sich Videoinstanzen der ausgewählten Videodateien erzeugen. Sofern in einem Matrixfeld ein Häkchen für die jeweilige Zuordnung von Gruppe und Videoinstanz gesetzt ist,

kann die betreffende Gruppe die jeweilige Instanz eines Videos einsehen. Das Konzept sieht weiterhin vor, innerhalb der Zellen die vorhandenen Features zu aktivieren oder zu deaktivieren. Aktivierte Features können somit für jede Zuordnung individuell konfiguriert werden.

### Zusammenfassung

In diesem Kapitel wurden videobasierte CSCL-Scripts begrifflich eingrenzt und als Schnittmenge videobasierter Scripts und CSCL-Scripts definiert. Anhand von Beispielen aus der Literatur sowie eigenen Feldversuchen lassen sich gegenwärtig drei Typen videobasierter CSCL-Scripts identifizieren: (1) Scripts zur Analyse und Diskussion von Videoinhalten, (2) Peer Annotation Scripts einschließlich dem Peer Assessment und (3) Jigsaw-Scripts, die das problembasierte Lernen mit Hilfe von Videos ermöglichen.

Auf Grundlage dieser Beispiele wurden strukturelle Bestandteile und Abläufe der Scripts differenziert beschrieben und im Hinblick auf eine Realisierung in einer CSCL-Umgebung spezifiziert. Die entstandene Spezifikation beantwortet die Forschungsfrage FF 2a. In der Spezifikation werden nicht nur die vielfältigen, theoretischen Ausprägungen deutlich, sondern auch die Besonderheiten, die sich aus dem Einsatz von kontinuierlichen Medien ergeben. Videos werden dabei nicht als monolithische Einheiten, sondern als zeitkontinuierliche Medien betrachtet, in denen weitere Informationen verankert sein können. Diese Sichtweise ermöglicht videospezifische Aufgaben und Aktivitäten, die in allgemeinen CSCL-Scripts keine Entsprechung finden und nur durch gesonderte Features im jeweiligen CSCL-System realisierbar sind. In Anlehnung an die *thinkLets* von Briggs und de Vreede [2009] werden divergente und konvergente Aktivitäten von solchen unterschieden, die die Organisation und Evaluation von Artefakten betreffen. Vor allem die divergenten Aktivitäten, wie etwa verschiedene zeitliche / räumliche Annotationen zum Zwecke der Analyse, Reflexion, Diskussion, Selektion oder allgemeinen Wissensgenerierung (vgl. S. 53), sind spezifisch für videobasierte CSCL-Scripts. Im vorliegenden Kapitel wurden 23 *thinkLets* beschrieben. Viele davon weisen einen direkten Bezug zu den Interaction Design Patterns für Videolernumgebungen auf, d.h. die dafür notwendigen technischen Lösungen sind bereits erprobt.

In Bezug auf die Formalisierung und Modellierung haben sich bestehende Ansätze und existierende CSCL-Systeme als ungeeignet erwiesen, die hier definierten videobasierten Scripts formal zu repräsentieren bzw. zu modellieren. Im Hinblick auf die gestellte Forschungsfrage FF 2b war es deshalb notwendig einen gesonderten Ansatz zu entwickeln, um die Scripts vollständig, skalierbar und konfigurierbar abzubilden. Auf Grundlage der Spezifikation und eines abgeleiteten *Entity-Relationship* Modells wurden erstens Teilnehmerinformationen einschließlich Gruppenformationen, zweitens Hypervideo-Dokumente inklusive der Nutzerbeiträge und drittens der Ablauf und die Konfiguration des eigentlichen Scripts logisch voneinander getrennt und gesondert in Datenstrukturen repräsentiert. Da es gegenwärtig jedoch noch keinen Standard für Hypervideo-Dokumente gibt und auch bestehende formale Sprachen für Scripts (insbesondere *IMS LD* und *S-BPM*) unzureichend sind, konnte eine Interoperabilität bzgl. anderer CSCL- oder Learning Management Systeme noch nicht erreicht werden.

Um generische videobasierte CSCL-Scripts auch modellieren zu können, wurde ein weitgehend formularbasierter Ansatz gewählt. In Anbetracht der Forschungsfrage FF 2c konnte ein



Workflow für eine grafische Benutzerschnittstelle konzipiert werden, der auf einfachen Eingabemaschinen sowie unterstützenden Daten- und Prozessvisualisierungen beruht. Um die Konfiguration eines Scripts insbesondere im Verlauf der Script-Phasen besser überblicken zu können, wurden grafische Hilfsmittel konzipiert. Inwieweit diese Konzepte sich in der Praxis bewähren, muss jedoch anhand eines realen CSCL-Systems, wie es im folgenden Kapitel vorgestellt wird, untersucht werden.

### Kernpunkte

- Unter videobasierten CSCL-Scripts versteht man die computerbasierte Form von Scripts, deren primäre Lernressourcen Videos darstellen.
- Hinsichtlich videobasierter Scripts unterscheidet man drei wesentliche Typen, die (1) zur Analyse und Diskussion von Videoinhalten dienen, (2) Mechanismen der Peer Annotationen und des Peer Assessment beinhalten und (3) das problembasierte Lernen mit Hilfe der Jigsaw-Methode abbilden.
- Ein videobasiertes CSCL-Script ist durch die Komponenten Teilnehmer, Gruppe, Gruppenformation, Lernressource, Software Feature und Aufgaben einschließlich der durch *thinkLets* beschreibbaren Aktivitäten spezifizierbar.
- Die Komplexität eines videobasierten Scripts erstreckt sich auf drei Level: (1) wenn voneinander abgegrenzte Phasen von Lernaktivitäten bestehen, (2) wenn zudem die Teilnehmer innerhalb von Kleingruppen Aufgaben bearbeiten (*intra-group*) und (3) wenn darüber hinaus Aufgaben auch gruppenübergreifend bearbeitet werden (*inter-group*).
- Im Datenmodell eines videobasierten Scripts sind neben dem Script-Verlauf teilnehmerbezogene Informationen in der Gruppenformation und Lernergebnisse in Hypervideo-Dokumenten repräsentiert.
- In Anbetracht der möglichen Komplexität videobasierter CSCL-Scripts wird für deren Modellierung ein Konzept einer formularbasiertes GUI vorgeschlagen.



## Kapitel 6

# Realisierung von Patterns und Scripts mit Hilfe von VI-TWO und VI-LAB

Die Ergebnisse der beiden vorangegangenen Kapitel waren eher theoretischer Art. Ohne weiteres lassen sich mit Hilfe der Interaction Design Patterns vielversprechende Videolernumgebungen entwerfen. Die sich daraus ergebenden Ansprüche an ein Software System sind jedoch sehr hoch. Die Analyse der 121 Videolernumgebungen hat jedenfalls gezeigt, dass keine der betrachteten Anwendungen alle Patterns instantiiert bzw. alle videobezogenen funktionalen Anforderungen erfüllt. Ähnlich verhält es sich mit den verschiedenen Typen videobasierter Scripts und den daraus abgeleiteten Spezifikationen: Für eine praktische Umsetzung fehlt es an geeigneten Werkzeugen. Insbesondere das Arrangement von strukturierten Gruppenlernprozessen anhand von Videos ist bislang mangels adäquater Werkzeuge nicht realisierbar.

In diesem Kapitel werden deshalb generische Hilfsmittel und Werkzeuge vorgestellt, die Lehrende, Interaction Designer und Entwickler bei der Konzeption und Entwicklung von Videolernumgebungen unterstützen. Die im Folgenden vorgestellten softwaretechnischen Lösungen leisten einen Beitrag für die weitere Erforschung von Lernprozessen anhand von interaktiven Videos.

Im ersten Teil des Kapitels geht es um *VI-TWO* als Framework für interaktive Videos, welches ein modulares Werkzeug für die Realisierung der Interaction Design Patterns darstellt und die Entwicklung vielfältiger Videolernumgebungen ermöglicht. Die unmittelbare Zielgruppe dieses Frameworks sind in erster Linie Software Entwickler, die beabsichtigen eine Videolernumgebung zu entwickeln oder zu erweitern.

*VI-LAB* als CSCL-System für videobasierte CSCL-Scripts steht im Mittelpunkt des zweiten Teils des Kapitels. In der Lernumgebung wurden die in Abschnitt 5.2 (S. 184) spezifizierten Komponenten und Mechanismen videobasierter CSCL-Scripts größtenteils realisiert. Mit Hilfe dieser Anwendung können Lehrende Scripts modellieren und ausführen. Möglich wird dies vor allem durch die Integration von *VI-TWO*, dessen Bestandteile (*Widgets*) je nach Bedarf für Gruppen von Lernenden aktiviert und konfiguriert werden können. Die Zielgruppe von *VI-LAB* bilden Lehrende, die individuelle oder kollaborative Lernszenarien mit Hilfe videobasierter Scripts konzipieren und durchführen möchten.

Der dritte Teil des Kapitels umfasst fünf beispielhafte Anwendungen der beiden zuvor genannten Systeme, welche bereits in der Praxis zum Einsatz gekommen sind. Zumal alle Anwendungen auf *VI-TWO* basieren, harmonisieren sie mit bestehenden Plattformen und Neuentwicklungen. Die Videolernumgebungen kamen im Kontext unterschiedlicher Fachbereiche zum Einsatz und waren bereits Gegenstand mehrerer Feldstudien. Die in Kapitel 4 identifizierten Interaction Design Patterns wurden dabei mehrfach instantiiert und praktisch erprobt.

## VI-TWO: JavaScript Framework für interaktive Videos

Bei dem im folgenden vorgestellten Software Framework *VI-TWO*<sup>15</sup> handelt es sich um generische Software Komponenten zur Realisierung der zuvor identifizierten Patterns im WWW. Das Framework wurde im Rahmen dieser Arbeit entwickelt. Es beruht auf der Vorlage der 45 Patterns und erlaubt die Umsetzung fast aller Patterns. Anders ausgedrückt lässt sich *VI-TWO* auf die einfache, an Calvo und Turani [2010] angelehnte Formel bringen:

$$\text{Framework} = (\text{Interaction Design Patterns} + \text{Software Komponenten})$$

Im Kontext dieser Arbeit werden Web Frameworks in Anlehnung an Vosloo und Kourie [2008] definiert und somit gegenüber Web-Anwendungen und Software Bibliotheken abgegrenzt:

**Definition 9.** *Ein Web Framework ist eine Kollektion von Software Komponenten, die die Entwicklung und Ausführung von webbasierten User Interfaces in einem Web-Browser unterstützen. Ein Framework besteht aus einem Kern von Komponenten, der um Widgets modular erweitert wird. Alle Komponenten sind durch Standardparameter definiert und hinsichtlich ihrer funktionalen und sichtbaren Gestalt flexibel konfigurierbar.*

Daraus ist die Definition eines Hypervideo Frameworks abgeleitet:

**Definition 10.** *Ein Hypervideo Framework ist eine plattformunabhängige Abstraktion generischer Funktionalitäten für die Entwicklung von Web-Anwendungen zur Wiedergabe, Navigation und Bearbeitung interaktiver, medial angereicherter und miteinander verknüpfter Videos.*

Durch die Implementierung sollte keinesfalls eine archetypische Videolernumgebung, sondern ein flexibles Werkzeug mit vielfältigen Gestaltungsmöglichkeiten für die Entwicklung von Videolernumgebungen geschaffen werden. Dies erschien notwendig, da sich viele Patterns nur in unzureichend zugänglichen oder inadäquaten Systemen wiederfinden, die für den Einsatz in der Lehre nicht verfügbar sind. Abgesehen von kostenpflichtigen Anwendungen wie *eduBREAK* betrifft das auch Forschungsprototypen wie *WEB DIVER*, historische Applikationen wie *MediaLoom*, aber auch Plattformen, die aufgrund der Datenschutzgesetze in Deutschland nicht in der Hochschullehre verwendet werden dürfen (z.B. *YouTube*). Ein weiterer Grund besteht darin, dass bestimmte Kombinationen bisher in keiner bekannten Anwendung zu finden sind. Ein Framework der hier beschriebenen Art ermöglicht es, beliebige Patterns als Komponenten miteinander zu kombinieren.

<sup>15</sup>VI-TWO ist ein Akronym für Videoplayer („VI“) der zweiten („TWO“) Generation im Sinne des Web 2.0.

Im Rahmen dieser Arbeit war es zudem notwendig eigene Erfahrungen mit der Implementierung und dem Einsatz von Videolernumgebungen zu sammeln, um die Patterns in der Praxis zu erproben, ihre Passgenauigkeit zu validieren und die Beschreibung der Lösungen zu verbessern. Überdies bildet das Framework eine wichtige Voraussetzung für die Modellierung und Untersuchung von videobasierten CSCL-Scripts (vgl. Abschnitt 6.2, S. 235).

Für die Community der Forscher im Bereich E-Learning, Pädagogik und Pädagogischer Psychologie soll das Framework nicht zuletzt die weitere Erforschung videobasierter Lernprozesse anregen. Das Framework soll helfen, neue Lernformate auf Grundlage innovativer Videolernumgebungen zu entwickeln. Es wurde deshalb versucht, die Anwendung von *VI-TWO* sehr einfach zu gestalten, um die Akzeptanz auch für Anwender mit nur grundlegenden Programmierkenntnissen zu erhöhen.

Nichtsdestotrotz eröffnen sich mit dem Framework Möglichkeiten, die Patterns in eigenständige Anwendungen zu überführen und selbige in vorhandene Systeme zu integrieren. Prinzipiell ergeben sich schon durch die Kombination von Patterns innerhalb von *VI-TWO* Grundgerüste für interessante Video(-lern)anwendungen und Webservices (vgl. Abschnitt 6.3, S. 246).

### 6.1.1 Anforderungen

Angesichts der eben angeführten Zielgruppenkonstellation ergeben sich funktionale Anforderungen für den Gebrauch der aus dem Framework hervorgehenden Produkte sowie technische Anforderungen für die Nutzung des Frameworks. Die funktionalen Anforderungen resultieren aus den Interaction Design Patterns unter Berücksichtigung der darin erläuterten Umsetzungsvarianten. Um jedoch den weitestgehend plattformunabhängigen Einsatz in Verbindung mit verschiedenen CMS/LMS zu gewährleisten, kann das Framework die Patterns nur soweit ermöglichen, wie clientseitig Operationen dies zulassen. Die Manipulation und Speicherung von Daten ist davon zunächst ausgeschlossen. Auf die Realisierung serverseitiger Komponenten wird jedoch im Zusammenhang mit CSCL-Scripts und *VI-LAB* (vgl. Abschnitt 6.2, S. 235) noch ausführlicher eingegangen.

Die in Tab. 6.1 dargestellten technischen Anforderungen für die Benutzung des Frameworks wurden aus der Literatur sowie aus den im nachfolgenden Teilabschnitt betrachteten Frameworks abgeleitet. Insbesondere in den Arbeiten von Lehner und Siegel [2009], Hardman [2004] und Cazenave et al. [2011] werden spezifische Anforderungen an Hypervideosysteme thematisiert, die für die Konzeption eines generischen Frameworks relevant sind.

Darüber hinausgehende Anforderungen in Bezug auf die Usability und *User Experience* sind nur marginal durch das Framework, vielmehr jedoch durch die damit implementierten und durch ein *User Interface* ausgestalteten Anwendungen bestimmt.

**Tabelle 6.1:** Übersicht der Anforderungen (R) an ein Framework für interaktive Videos**Medienintegration**

R1	Anreicherung und Integration von kontinuierlichen und diskreten Medien sowie dynamischen Inhalte aus dem WWW	[Hofmann et al. 2009]
R2	Skalierbare synchrone Darstellung verschiedener Medien und Annotationen	[Cazenave et al. 2011; Lehner und Siegel 2009]
R3	Lineare Navigation innerhalb eines Videos	
R4	Zeitbezogene, nicht-lineare Navigation innerhalb der medialen Gesamtstruktur	[Cazenave et al. 2011]
R5	Komposition von mehreren Videos als zeitliche Sequenz einzelner Medien	[Lehner und Siegel 2009]

**Zugänglichkeit**

R6	Anreicherung der Lernressourcen mit semantischen Informationen, auch im Hinblick auf SEO	[Hardman 2004]
R7	Unterstützung der <i>Media Fragment URI 1.0</i> hinsichtlich der Adressierung zeitlicher und namentlicher Fragmente	[Hausenblas et al. 2011]
R8	Segmentierung von Videoinhalten	[Hardman 2004]
R9	Berücksichtigung von Standards zur Barrierefreiheit	[Caldwell et al. 2008]
R10	Unterstützung der Mehrsprachigkeit	[Lehner und Siegel 2009]

**Produktion von Hypervideos**

R11	Unterstützung kollaborativer Produktionsszenarien, etwa durch synchrone Kommunikation und Produktion sowie eine Versionierung	[Hofmann et al. 2009]
R12	Skalierbarer Ansatz zur Annotationen von Videos mit Hilfe geeigneter Autorenwerkzeuge	[Hardman 2004; Lehner und Siegel 2009]
R13	Import / Export von Medien und Daten zur Nutzung in anderen Systemen (z.B. CMS, LMS)	[Hofmann et al. 2009; Lehner und Siegel 2009]

**Anwendbarkeit für Software Entwickler**

R14	Deklarative Auszeichnung des spatialen und temporalen Layouts, die auch Interaktionselementen Rechnung trägt	[Cazenave et al. 2011]
R15	Trennung von Inhalt, Layout und Steuerung	[Cazenave et al. 2011]

(Fortsetzung auf der nächsten Seite)

(Tabelle 6.1 – Fortsetzung von der vorherigen Seite)

R16	Weitgehend geräteunabhängige Implementierung auf Basis von offenen Webstandards, einschließlich einer nativen Darstellung im Browser	[Cazenave et al. 2011]
R17	Dokumente oder Datenaustauschformate wie LD, SCORM gewährleisten die Interoperabilität mit anderen Autorenumgebungen und Abspielkomponenten	vgl. Teilabschnitt 3.4.2 (S. 86)
R18	Modularität und Erweiterbarkeit des Frameworks	[Lehner und Siegel 2009]

Die Nummerierung der Anforderungen (R1...18) wird im Verlauf der folgenden Seiten diejenigen Textabschnitte kennzeichnen, in denen die Umsetzung der jeweiligen Anforderungen beschrieben ist.

### 6.1.2 Verwandte Arbeiten

Im Vergleich zur Anzahl an Videolernumgebungen gibt es nur wenige generische Frameworks für die individuelle Entwicklung von Videolernumgebungen. Folgende sechs Anwendungen konnten hinsichtlich der zuvor vereinbarten Anforderungen miteinander verglichen werden.

- **popcorn.js:** Das von der *Mozilla Foundation* entwickelte JavaScript Framework zielt darauf, HTML5-Videos (und auch Audio) mit Informationen im Web zu verknüpfen, diese zu integrieren und zu synchronisieren. Das Framework ist durch Plugins erweiterbar. Plugins steuern die Informationsdarstellung in Abhängigkeit der Wiedergabezeit. In welcher Weise die Darstellung erfolgen soll, ist in JSON-Objekten gemäß einem einheitlichen Schema definiert. Mit dem Webservice und dem frei verfügbaren Werkzeug *Mozilla Popcorn Maker* lassen sich diese Hypervideo Dokumente erstellen und bearbeiten [Mozilla-Foundation 2012].
- **timesheet.js:** Das Framework erweitert HTML5 und CSS3 um deklarative Auszeichnung für zeitliche Ereignisse in Multimedia Dokumenten [Cazenave et al. 2011]. Im so genannten *timeContainer* sind Elemente zusammengefasst, die entweder in zeitlicher Relation zueinander oder in Bezug zur Zeitachse eines Videos als zeitliche Ereignisse arrangiert sind. Auf diese Weise lassen sich auch Medien miteinander synchronisieren. Der Vorteil dieses Ansatzes besteht darin, die zeitliche und logische Struktur in einem Dokument zu vereinen.
- **Flowplayer:** Dieser Player stellt kein vollständiges Framework dar, bietet jedoch eine JavaScript API, mit der so genannte *cuepoints* mit Untertiteln, *overlays*, Hyperlinks und anderen zeitabhängigen Annotationen assoziiert werden können.
- **cacophony.js:** Als experimenteller Videoplayer bietet cacophony ein rudimentäres Framework, welches ansatzweise die Integration von Überprüfungsfragen sowie Hyperlink-Verzweigungen ermöglicht.
- **Projektor:** Der Videoplayer beinhaltet ein rudimentäres Framework, welches sich durch eine optimierte Auslieferung von Videoinhalten in Anbetracht der Übertragungskapazitäten und Displayverhältnisse auszeichnet. Mangels einer Dokumentation konnten viele



Funktionen, wie z.B. die Playlist, dieses kommerziellen Angebots nicht in die Untersuchung einbezogen werden.

- **YouTube:** Mit Hilfe verschiedener APIs<sup>16</sup> (*Player*, *IFrame*, *Data* und *Analytics API*) kann man *YouTube* auch als ein Software Framework für den Zugang zu einer sehr umfangreichen Datenbasis verstehen. Die Entwicklung individueller Anwendungen wird ermöglicht. Die APIs erlauben neben der Datenverarbeitung auch die Konfiguration des Videoplayers. Die Manipulation und Annotation von Videos ist jedoch nur auf den Upload von Videos und die Bearbeitung von Metadaten beschränkt. Auch die Funktionalität des Players ist nicht im vollen Umfang mit Hilfe der APIs konfigurierbar, d.h. das Portal von *YouTube* bietet mehr Interaktionsmöglichkeiten als Entwickler mit Hilfe der APIs in externen Anwendungen realisieren können. Oftmals lassen sich Funktionen lediglich an- und abstellen (z.B. ähnliche Videos, Suche), ohne dass ein Einfluss auf die Datenverarbeitung und -repräsentation genommen werden kann.

In Tab. 6.2 sind die Frameworks den Anforderungen des vorherigen Teilabschnitts sowie der Anzahl an realisierten Patterns gegenübergestellt. In Anbetracht der Patterns, welche die Mikrointeraktivität beschreiben, deckt keines der betrachteten Frameworks die volle Bandbreite an bewährten Lösungen ab. *Popcorn.js* deckt immerhin 15 der 29 Patterns ab. Die modulare Erweiterbarkeit durch eine Plugin-Architektur sowie die Distribution als *Open Source Software* fanden in der Community der Entwickler im Umfeld der *Mozilla Foundation* eine breite Resonanz. In der Folge entstanden sehr vielfältige Plugins. Wie *popcorn.js* fokussieren die meisten dieser Frameworks jedoch nur die Mikrointeraktivität und vernachlässigen semantische und strukturelle Verknüpfungen von Videos innerhalb einer Kollektionen. *YouTube* wiederum bietet ein umfangreiches Set für Interaktionen innerhalb der medialen Gesamtstruktur. Im Sinne der Definition eines Frameworks handelt es sich bei den APIs jedoch nur um Schnittstellen, welche nur mit den auf *YouTube* veröffentlichten Videos nutzbar sind. Neben der flexiblen Integration von Daten und Lernressourcen ist auch die Erweiterbarkeit stark eingeschränkt. Zudem werden in den Nutzungsbedingungen bestimmte Nutzungen und Zwecke ausgeschlossen<sup>17</sup>.

<sup>16</sup>Siehe [https://developers.google.com/youtube/player\\_parameters](https://developers.google.com/youtube/player_parameters) (angerufen am 1.08.2015).

<sup>17</sup>Siehe <https://developers.google.com/youtube/terms?hl=en> (abgerufen am 28.09.2015).

**Tabelle 6.2:** Gegenüberstellung verwandter Frameworks mit den 18 Anforderungen (● Anforderung erfüllt, ○ teilweise erfüllt, – nicht erfüllt) sowie der Anzahl umgesetzter Patterns auf Mikro- und Makroebene der Interaktivität.

	popcorn.js	timesheet.js	Flowplayer	cacophony.js	Projektor	YouTube
R1	●	●	●	?	○	○
R2	●	–	●	?	–	–
R3	●	●	●	●	●	●
R4	●	–	–	○	–	○
R5	●	–	–	●	–	●
R6	?	–	–	–	–	●
R7	○	○	○	–	–	○
R8	●	●	●	–	–	–
R9	○	○	○	–	○	●
R10	–	–	●	–	○	●
R11	–	–	–	–	–	–
R12	●	–	–	–	–	●
R13	●	○	–	–	–	○
R14	–	●	–	–	–	–
R15	●	●	●	○	?	●
R16	●	●	●	–	●	●
R17	○	–	–	–	–	–
R18	●	●	●	–	–	–
Patterns bzgl. Mikrointeraktivität	>15	>5	>4	3	2	9
Patterns bzgl. Makrointeraktivität	3	0	0	1	0	11

### 6.1.3 Architektur von VI-TWO

**R16** Zur Ausführung von Programmcode gibt es auf Seiten des Web-Clients mit ECMAScript [W3C 2011b] nur einen relevanten Standard [Vosloo und Kourie 2008]. Das gesamte Framework ist deshalb in JavaScript implementiert. Es setzt jedoch auf anderen Bibliotheken wie *jQuery* oder *jQueryUI* auf, um im Ergebnis eine auf HTML5 und CSS2 basierende Web-Anwendung bereitzustellen. Die genannten Bibliotheken deuten bereits an, dass es sich um eine clientseitige Anwendung handelt. Prinzipiell können damit Anwendungen ohne eine Serveranbindung realisiert werden (vgl. *IWRM education*, S. 247), wobei dabei eine Speicherung von nutzerseitigen Daten ausgeschlossen ist. Die serverunabhängige Implementierung eröffnet jedoch die Möglichkeit *VI-TWO* in verschiedene Serveranwendungen, insbesondere LMS oder CMS, zu integrieren, da lediglich die Datenschnittstellen für das Hypervideo-Dokument (i.d.R. Metadaten, Video-URI, Annotationen) definiert werden müssen.

Grundlegend besteht das Framework aus (1) einem Player für die Videowiedergabe, (2) einem Observer zum Erfassen und Weiterleiten von Zustandsänderungen und (3) einer Clock zur Überwachung zeitlicher Ereignisse. Dieser Kern lässt sich flexibel erweitern. Diese Erweiterungen werden als Widgets bezeichnet, die sich in drei Kategorien einteilen lassen:

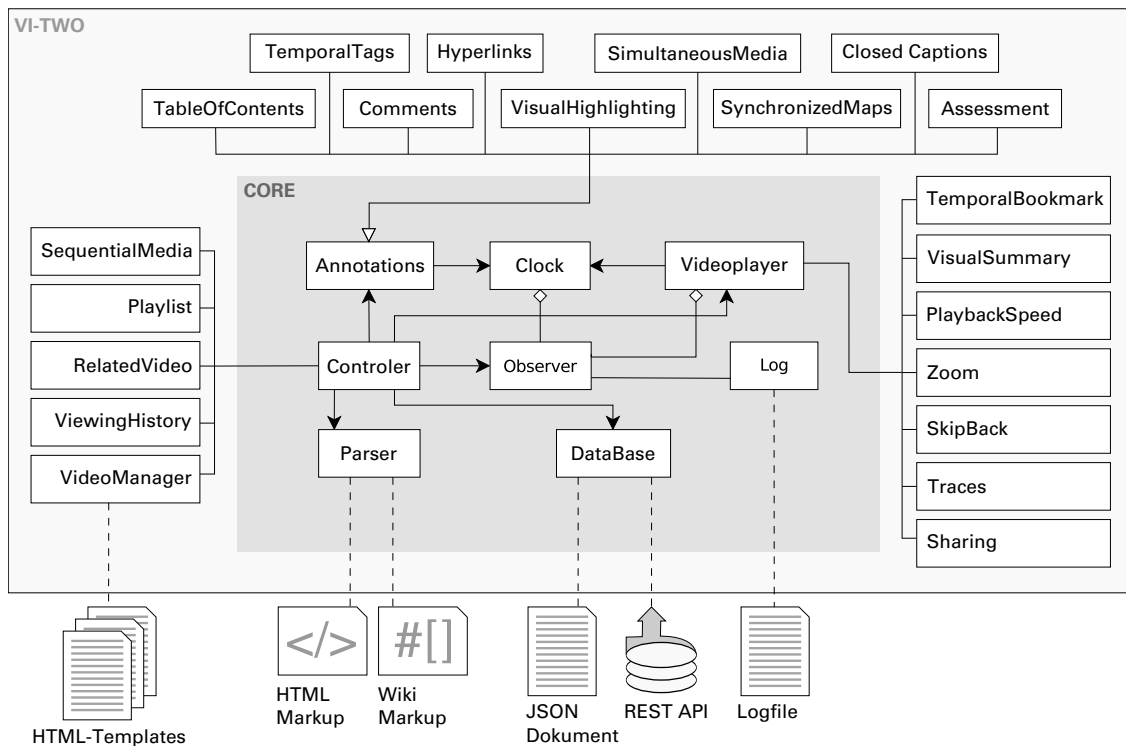
- Videoplayer (z.B. Wiedergabegeschwindigkeit, Zoom)
- Videoannotationen (z.B. Kapitelmarken, Überprüfungsaufgaben)
- Kollektionen von Videos (z.B. ähnliche Videos, Suche)

Der Videoplayer stellt zur Steuerung der Videowiedergabe Schnittstellen für den *Observer* bereit. Der *Observer* fragt bei Bedarf Systemzustände, Ereignisse und Daten ab (*pull notification*), um sie an Widgets weiterzuleiten. Umkehrt werden Zustandsänderungen und Ereignisse einschließlich der veränderten Daten an den *Observer* geschickt, der diese Information an bestimmte Widgets weiterleitet (*push-update notification*). Die Widgets sind dabei vom Player und dessen Schnittstellen unabhängig und kommunizieren nur mit dem Observer.

Die Wiedergabezeit eines Videos überwacht ein gesonderter Observer namens Clock. In kurzen Zeitintervallen prüft die Clock, ob eine zeitabhängige Annotation mit der aktuellen Wiedergabezeit in Zusammenhang steht. Falls ja, löst sie einen *Event* aus, den das entsprechende Widget behandelt. Auf diese Weise werden die Informationen im *push*-Verfahren an die Widgets kommuniziert. Zeitbasierte Ereignisse lassen sich somit in HTML-Dokumenten mittels JavaScript simulieren. Details dieses Verfahrens werden im Anschluss auf S. 227 erläutert.

Die einzelnen Komponenten des Frameworks sind syntaktisch als Klassen mit Vererbungsbeziehungen organisiert, wobei im Hintergrund nur prototypbasierte Objekte miteinander verknüpft sind (*prototype chains*). Entlang dieser Ketten werden zur Laufzeit der Anwendung die Eigenschaften eines Objekts vom Prototyp-Objekt kopiert, d.h. scheinbar „vererbt“. Trotz dieser klassenlosen Objektorientierung sollen die Beziehungen der Komponenten in Abb. 6.1 zum besseren Verständnis im Stile eines Klassendiagramms dargestellt werden.

*Events* stellen das zentrale Element der Kommunikation zwischen einzelnen Objekten und Komponenten innerhalb von *VI-TWO* dar. Objekte bilden dabei Quellen für spontan auftretende Ereignisse, wie das Laden eines Videos, das Auftreten einer zeitabhängigen Annotation nach der Navigation auf der Zeitleiste oder das Ende eines Videos. Andere Objekte können de-



**Abbildung 6.1:** Architektur von *VI-TWO* mit dem Kern (Core) bestehend aus Videoplayer, Observer, Clock, Log und abstrakten Annotations. Um den Kern herum sind die drei Typen von Widgets angeordnet: Annotations (oben), Player-Widgets (rechts) und die Widgets für die Makrointeraktivität (links).

finierte *Events* beobachten und beim Eintreten entsprechend behandeln, indem zum Beispiel die Wiedergabe des Videos beginnt, ein Bildelement angezeigt oder das nächste Video geladen wird.

*VI-TWO* ermöglicht es, unterschiedliche Daten wie Videos, Bilder, Karten und Webseiten aus verschiedenen Quellen im Web in einer einzigen Repräsentation zusammenzuführen. Durch diese Kombination von Daten entsteht ein *data mashup* als eigenständiger Webservice. **R1**

Um die Benutzung möglichst ohne Verzögerungen beim Laden und beim Seitenaufbau gewährleisten zu können, operiert *VI-TWO* als so genannte *Single Page Web-Application*. Der grundlegende HTML-Code wird dabei schon bei der ersten HTTP-Anfrage ausgeliefert. Alle folgenden Anfragen erfolgen asynchron über eine *REST API*, lokale Dateien auf dem Server (z.B. **R18** HTML-Templates) oder sonstige Web-Ressourcen.

## Widgets

Die Modularität des Frameworks ist durch *Widgets* verwirklicht, die den Kern des Frameworks jeweils um bestimmte Funktionalitäten erweitern. Jedes Widget nutzt Kernkomponenten des Frameworks und ist beliebig mit anderen Widgets kombinierbar. Abhängigkeiten zwischen den Widgets wurden weitestgehend vermieden. Die Repräsentation und Funktionalität eines Widgets ist durch Standard-Parameter (*options*) definiert. Diese können jedoch bei der Initialisierung gemäß individueller Anforderungen überschrieben werden. Die Trennung von Inhalt, Darstellung und Logik ist ein zentraler Aspekt des Frameworks. **R15**

stellung und Steuerung wurde je Widget nach dem Pattern *model-view-controller* [Gamma et al. 1995] clientseitig realisiert. Die zu verarbeitenden Datenmodelle werden durch die Kernkomponenten als HTML DOM Elemente (teilweise als Template) bereitgestellt, während die Angaben zur Darstellung (*view*) in einem gesonderten *Cascading Style Sheet* (CSS) gekapselt sind. Die eigentlichen Widgets in Form von JavaScript-Klassen enthalten nur die Logik zur Ausführung, Manipulation und Steuerung der *view* und *model* Komponenten. Die Kommunikation mit den Kernkomponenten, insbesondere dem Videoplayer, erfolgt ausschließlich über Schnittstellen zum *Observer*. Durch diese Form der Modularisierung und funktionalen Trennung lässt sich das Framework mit geringem Aufwand um zusätzliche Widgets erweitern und flexibel anpassen (vgl. „hello world“ Widget im Anhang A.17, S. 342). Die Modularisierung ermöglicht außerdem, dass wirklich nur die erforderlichen Widgets zur Laufzeit geladen werden müssen.

### 6.1.4 Videoplayer

- R1** Der Videoplayer basiert auf HTML5 und ermöglicht die Wiedergabe gängiger Video-Codecs, je nach verwendeten Browser. Die Videos referenziert der Player von lokalen Dateien oder Quellen im Web, einschließlich Diensten wie *YouTube*.

Aus Anwendersicht besteht der Player aus dem Videobild und elementaren Steuerungselementen (siehe BASIC CONTROLS). Das Videobild skaliert je nach Größe des Fensters oder Displays und kann sich auf die Größe des Fensters, bis hin zum echten Browser-Vollbild ausdehnen (siehe APPROPRIATE DELIVERY und FULL SCREEN). Sollte sich die Wiedergabe, etwa aufgrund von Übertragungsengpässen, verzögern, wird dies dem Anwender sichtbar gemacht (siehe LOADING INDICATOR). Zur Steuerung des Videos stehen mindestens Schaltflächen für Play und Pause sowie Schieberegler für die Navigation auf der Zeitleiste und zur Regelung der Lautstärke zur Verfügung (siehe BASIC CONTROLS). Auch Tastenkombinationen für eine einfachere und barrierefreie Bedienung werden dabei im Observer registriert und weitergeleitet (siehe SHORTCUT COMMANDS). Darüber hinaus sind eine Reihe spezifischer Player-Widgets verfügbar:

- *vi2.annotated-timeline*: Schnittstelle für andere Widgets, um zeitliche Ereignisse auf definierten Spuren der Zeitleiste abzubilden (siehe ANNOTATED TIMELINE).
- *vi2.traces*: Analysiert erhobene Logdaten der Videonutzung und stellt diese in einer *heatmap* parallel zur Zeitleiste dar (siehe USER TRACES).
- *vi2.skip-back*: Rücksprung um wenige Sekunden, um die Wiedergabe der letzten Worte noch einmal zu wiederholen (siehe SKIP BACK).
- R9** – *vi2.playback-speed*: Stufenweise Anpassung der Wiedergabegeschwindigkeit (siehe PLAYBACK SPEED).
- R9** – *vi2.zoom*: Zoomfunktion für das Bewegtbild (siehe ZOOM).
- *vi2.sharing*: Bedienelement, welches zur Generierung eines HTML-Codes genutzt werden kann, um das Hypervideo in eine andere Webseite einbetten zu können.
- *vi.temporal-bookmarks*: Um die Position der Videowiedergabe referenzieren oder einfach nur abspeichern zu können, bietet dieses Widget eine komfortable Möglichkeit (siehe TEMPORAL BOOKMARKS).
- *vi.scrubber*: Vorschau der Videoszenen beim Überstreichen der Zeitleiste (siehe VISUAL SUMMARY)

### 6.1.5 Videoannotationen

Videoannotationen erlauben es, ein Video mit zusätzlichen Informationen anzureichern und zusätzliche Interaktionsmöglichkeiten zeitlich und räumlich zu integrieren. Annotationen sind dabei nicht auf bestimmte Medien beschränkt. Neben Texten und Bildern können ebenso Animationen, andere Videos sowie dynamische Inhalte von Webseiten, einschließlich Kartendiensten, mit dem Video assoziiert sein. Weit verbreitet sind die mit dem Video synchronisierten Präsentationsfolien und zeitlich verankerten Kommentare.

Mit Verweis auf den bereits beschriebenen Gestaltungsraum von Hypervideos (vgl. Teilabschnitt 3.2.1, S. 70) kann es sich bei einer Annotation um ein zeitbezogenes oder ein zeitabhängiges Objekt handeln. Zeitbezogene Objekte referenzieren einen Zeitpunkt oder einen Zeitabschnitt im Video. Typische Vertreter sind die Elemente eines Inhaltsverzeichnisses sowie zeitgenaue Schlüsselwörter und Suchergebnisse. Zeitabhängige Objekte werden in Abhängigkeit der Wiedergabezeit angezeigt und können aus Bildern, synchronisierten Videos oder Web-Seiten, aber auch aus Hyperlinks bestehen. Letztere beziehen sich in ihrer Funktion wiederum auf einen Zeitpunkt in einem Video oder eine externe Ressource. Darüber hinaus sind Annotationen zumeist innerhalb oder außerhalb des Videobildes platziert. Abb. 6.2 stellt eine Klassifikation der in *VI-TWO* realisierten Annotationen dar.

		Zeitliches Layout	
		zeitbezogen	zeitabhängig
Räumliches Layout	Innerhalb des Videos	hyperlinks branching video detail on demand	overlays closed captions visual highlighting  assessment
	Außerhalb des Videos	table of content temporal tags	comments transcript user notes  simultaneous media synchronized map

**Abbildung 6.2:** Klassifikation von Videoannotationen hinsichtlich des zeitlichen und räumlichen Layouts

## Repräsentation und Datenverarbeitung von Annotationen in Hypervideo-Dokumenten

Ein Hypervideo-Dokument besteht in *VI-TWO* aus einem um *Custom Elements* erweiterten **R14** HTML-Dokument. Das Standard-Set an HTML-Auszeichnungen wird somit um *Document Object Model*-Elemente für zeitliche und räumliche Arrangements von Videos und Videoannotationen erweitert. Je nach Art der Annotation und des Anwendungskontexts sind verschiedene Datenattribute angegeben, welche neben Metadaten (z.B. Autor, Erstellungszeitpunkt), die zeitliche und räumliche Platzierung sowie den Inhalt (z.B. Text-Label, Inhalt eines Kommentars, Verweisziel) enthalten. Im Anhang A.18 (S. 343) findet sich eine detaillierte Aufstellung der definierten Elemente und Attribute.

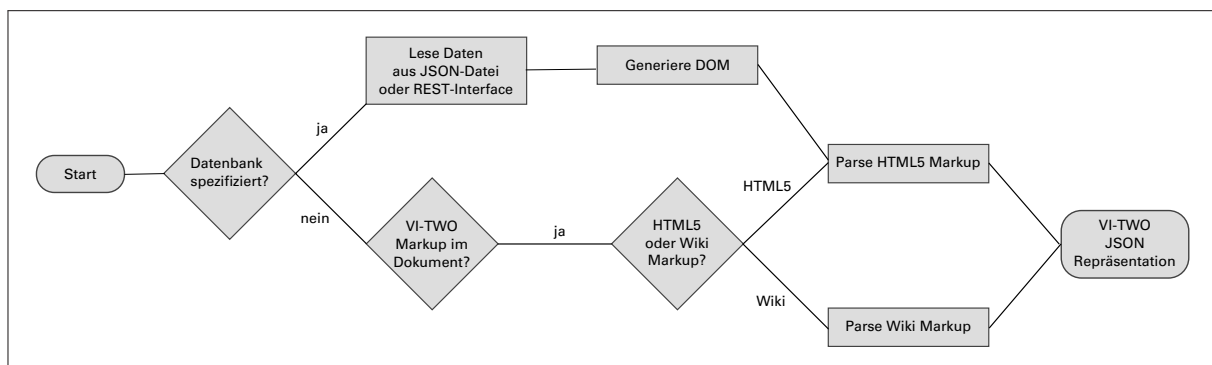
Ein Hypervideo-Dokument ist zur Laufzeit entweder bereits im HTML-Dokument enthalten oder es wird dynamisch aus einer Datei bzw. einem REST-Interface als JSON-Dokument geladen und entsprechend umgewandelt. Daraus ergeben sich vier Möglichkeiten, um ein Dokument zu laden:

- asynchron von einer Datei geladenes JSON-Dokument (z.B. `hypervideo-daten.json`);
- asynchron von einem REST-Interface bereitgestelltes JSON-Dokument, z.B.  
`http://www.example.com/getCommentsOfVideo/123`;
- statisch im HTML-Dokument enthaltenes DOM-Element, z.B.  

```
<vi2-video data-scr="video-datei.mp4">
<vi2-comment data-start=45.4 data-duration=10 data-content="Ein Kommentar">;
```
- statisch im HTML-Dokument als Zeichenkette eines DOM-Elements enthaltene *Wiki Markup*.

Unabhängig von der Datenquelle werden damit deklarative Auszeichnungen im HTML-Dokument verankert. Obwohl das Datenmodell intern als JSON repräsentiert und verarbeitet wird, erleichtern die deklarativen Auszeichnungen die Pflege und Wiederverwendung des gesamten Dokuments. Somit ist es beispielsweise möglich, eine lokal gespeicherte Webseite mit einem *VI-TWO* Video auch ohne Verbindung zu einem Server auszuführen und zu bearbeiten. In Abb. 6.3 ist die Datenverarbeitung von Hypervideo-Dokumenten zusammenfassend in einem Flussdiagramm veranschaulicht. Die JSON-Repräsentation erlaubt demgegenüber den Export

**R13** und Import der Dokumentdaten.



**Abbildung 6.3:** Fließdiagramm der Verarbeitung deklarativer Auszeichnungen für die Repräsentation innerhalb von *VI-TWO*.

## Zeitbezogene Annotationen

Zeitbezogene Annotationen dienen vorrangig einer verbesserten Navigation innerhalb des zeitlichen Layouts bzw. innerhalb der zeitlich determinierten Sequenz von audiovisuellen Informationen im Video. In den folgenden Widgets wurden zeitbezogene Annotationen realisiert:

- R8** – *vi2.TableOfContents*: Eine chronologisch nach dem Auftreten im Video geordnete Liste von Bezeichnungen (z.B. Kapitel, Sektionen, Sequenzen) für inhaltlich zusammengehörige Abschnitte im Video.

- R8, R6** – *vi2.Tags*: Einem Video können mehrere Schlüsselworte zeitgenau zugeordnet werden. Die Auswahl an verfügbaren Schlüsselworten kann dabei auf eine Menge reduziert oder



vollkommen frei gestaltet sein. Im Unterschied zu den Kapitelmarken, kann ein Schlüsselwort mit mehreren zeitlichen Positionen verknüpft sein. Die Darstellung erfolgt i.d.R. als alphabetisch geordnete Liste oder in Form einer so genannten *tag cloud*.

- *vi2.Comments*: zeitlich und räumlich im Video verankerte Kommentare.
- *vi2.Transcript*: Schriftliche Darstellung der im Video gesprochenen Worte und Dialoge. **R9**  
Während der Wiedergabe werden die korrespondierenden Abschnitte im Transkripttext optisch hervorgehoben.

## Zeitabhängige Annotationen

Anstatt Zusatzinformationen wie Untertitel, Hervorhebungen oder simultane Medienströme produktionsseitig im Video zu kodieren, erfolgt in *VI-TWO* eine getrennte, jedoch abgestimmte Verarbeitung zeitabhängiger Annotationen (siehe *OVERLAYS*). Die Sichtbarkeit einer Annotation kann in Abhängigkeit der Wiedergabezeit definiert werden, falls dadurch eine semantische Beziehung zwischen den audiovisuellen Inhalten des Videos und der Annotation ausgedrückt werden soll. Beispielsweise sollte eine Präsentationsfolie genau dann neben dem Video angezeigt werden, wenn der Sprecher im Video dazu spricht. Wie präzise die Synchronisierung erfolgen muss, hängt von der Art der Annotation ab. Während Untertitel (*CLOSED CAPTIONS*) auf eine halbe Sekunde präzise synchronisiert werden sollten, besteht bei zeitgenauen Überprüfungsfragen (*ASSESSMENT*) ein Toleranzbereich von einer Sekunde.

In den folgenden Widgets wurden zeitabhängige Annotationen realisiert:

- *vi2.SynchronizedImages*: Bilddateien werden ab einem definierten Zeitpunkt so lange **R1**  
eingebildet, bis die nächste Bilddatei in Abhängigkeit der Wiedergabezeit die dafür vorgesehene Fläche einnimmt.
- *vi2.SynchronizedMaps*: Zu einem gegebenen Zeitpunkt werden Koordinaten oder Polygonen von Koordinaten auf einer Karte (hier *OpenStreetMap*) neben dem Video dargestellt.
- *vi2.VisualHighlight*: Zu einem gegebenen Zeitpunkt werden Bereiche im Videobild durch eine farbige, halbtransparente Fläche oder einen Rahmen für eine definierte Zeitdauer hervorgehoben.
- *vi2.Hyperlinks*: (siehe nächste Seite)
- *vi2.Assessment*: Zu einem Zeitpunkt werden Freitext- oder Multiple Choice Aufgaben eingebildet. Ferner können auch Lückentextaufgaben über den dargestellten Text in synchronisierten Präsentationsfolien gelegt werden (vgl. Seidel und Jödicke [2014]).
- *vi2.ClosedCaptions*: Wahlweise können Untertitel im unteren Bildrand eingebildet werden. Falls die Untertitel in mehreren Sprachen vorliegen, können Anwender eine Auswahl **R9**  
treffen. Unterstützt werden die Formate *SubRip*, *W3C WEBVTT* und *W3C TTML*. **R10**

Ferner werden auch in den Widgets für zeitbezogene Annotationen zeitabhängige Ereignisse berücksichtigt. Beispielsweise wird das jeweils aktuelle Kapitel im Inhaltsverzeichnis (*vi2.TableOfContents*) gegenüber den anderen hervorgehoben.

Der in Abschnitt 3.3.2 (S. 78) bereits thematisierte lineare Aufwand bei der Ermittlung der **R12**  
ein- oder auszubblendenden Annotationen innerhalb möglichst kleiner Zeitintervalle wird durch den Algorithmus 2 verbessert. Anstatt innerhalb eines Intervalls alle Annotationen zu prüfen, **R2**

werden hierbei nur jene geprüft, die in Relation zur aktuellen Minute der Medienzeit stehen. Dazu werden die Annotationen in einen invertierten Index der zeitlichen Positionen überführt. Dieser Index wird anschließend gemäß der Zeiteinheiten von Stunden und Minuten in einen zweistufigen Index verschachtelt. Zu einem gegebenen Zeitpunkt  $t$  während der Wiedergabezeit werden nur jene Annotationen überprüft, die dem Index  $I_{Stunde(t),Minute(t)}$  zugeordnet sind. Die asymptotische obere Grenze der indexbasierten Suche nach Annotationen pro Zeitintervall entspricht  $\log(n)$ .

**Input :**

$A$  Liste der Annotationen eines Videos

$A_{id}$  Index der Annotationen

$t$  aktuelle Abspielposition des Videos

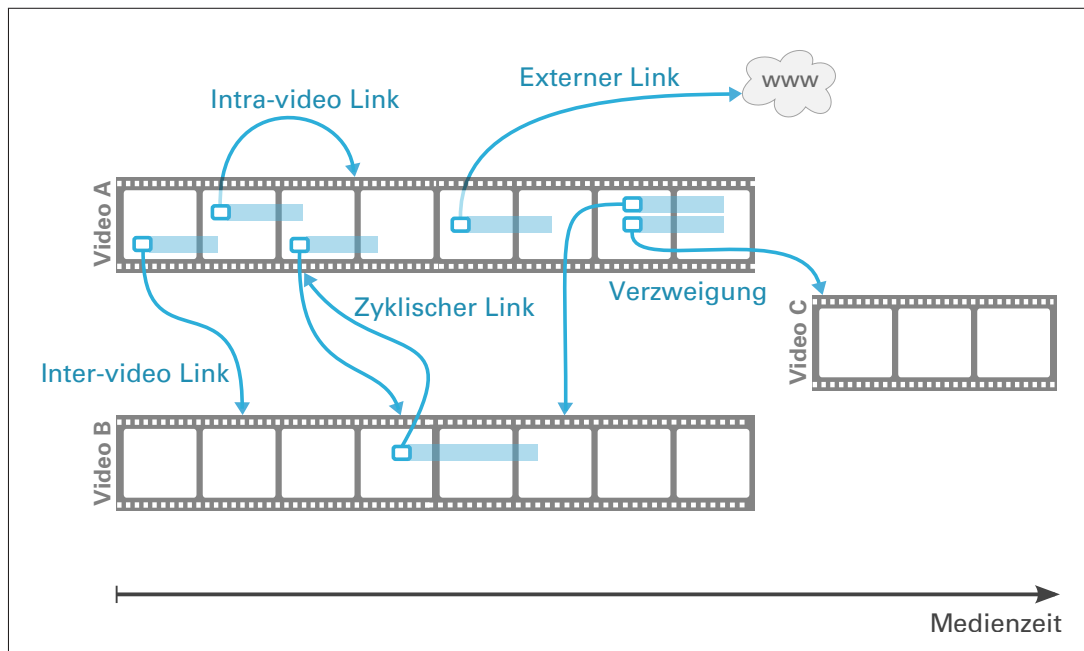
**Function** displayAnnotation()

```

    index  $\leftarrow$  getSegmentOfPlaybackTime( $t$ )
    for  $a \in A_{id}[index]$  do
        if  $a.start \leq t \leq a.end$  then
            | show( $a$ )
        else
            | hide( $a$ )
        end
    end
    end
    return
End

```

**Algorithmus 2 :** Indexbasierte Suche nach Annotationen, die in Abhängigkeit der Wiedergabezeit ein- oder ausgeblendet werden



**Abbildung 6.4:** Schematische Darstellung der im Widget *vi2.Hyperlinks* innerhalb von *VI-TWO* implementierten Hyperlink-Varianten: Zyklischer Link (DETAIL ON DEMAND), Verzweigung (BRANCHING VIDEO) sowie externer, intra-video und inter-video Link (HYPERLINKS)

## Hyperlinks

Hyperlinks erscheinen in Abhängigkeit der Zeit. Ihr Verweisziel kann einen zeitlichen Bezug zur Medienzeit des aktuellen oder eines anderen Videos haben. In Abb. 6.4 sind alle von *VI-TWO* unterstützen Varianten von Hyperlinks skizziert. Das Pattern **HYPERLINK** sowie die beiden Sonderfälle **BRANCHING VIDEOS** und **DETAIL ON DEMAND** wurden in einem Widget zusammengefasst. Der Videoplayer erfüllt annähernd die Anforderungen eines von Meixner und Hoffmann [2012, S. 30] beschriebenen „real non-linear player[s]“. Möglich wurde dies durch die Adressierung von einzelnen Medienelementen, statt der vollständigen Seite. Die Latenz ist jedoch von der Übertragungskapazität der Internetverbindung, dem verwendeten Web-Browser, der Dateigröße des Videos sowie vom adressierten Bereich innerhalb des Videos abhängig. **R4**

### 6.1.6 Makrointeraktivität in Kollektionen von Videos

*VI-TWO* stellt nicht nur einen erweiterten Videoplayer dar, sondern ist auch ein Werkzeug für die Handhabung umfangreicher Kollektionen von Videos. Das Stöbern, Suchen sowie die Organisation der Ressourcen wird in einer Reihe von Patterns adressiert, deren korrespondierende Software Komponenten in Form von Widgets im Folgenden vorgestellt werden.

## Video Manager

Als eine der zentralen Komponenten gewährt der Video Manager eine Übersicht über die zu einer Auswahl von Videos gehörenden Metainformationen. Diese Darstellung sollen dem Anwender helfen, sich einen Überblick zu verschaffen und ein einzelnes Video für die Wiedergabe auszuwählen. Welche Videos nach welchen Kriterien ausgegeben werden, ist vorrangig durch die jeweiligen Widgets bestimmt. Ein Widget registriert dazu im Video Manager Fragmentbezeichner als Präfix der URL, sodass bestimmte Ansichten bzw. Informationen eindeutig referenzierbar und innerhalb der Navigationshistorie des Browsers navigierbar werden (siehe **JOURNALED NAVIGATION** und **VIEWING HISTORY**). Wird eine solche Route via HTTP im Browser aufgerufen, ermöglicht der Video Manager, dass die vom Widget übergebenen Daten in einem definierten HTML-Template dargestellt werden. Darüber hinaus sind Standard-Routen, wie eine Übersicht aller Videos, im Video Manager definiert. In der nachfolgenden Auflistung sind Fragmentbezeichner für exemplarische Routen aufgeführt:

```
#!/videos/all – Übersicht über alle Videos;
#!/category/:category – Übersicht aller Videos in der Kategorie category (z.B. „Repetitorium“);
#!/video/:id – Aufruf eines durch seine eindeutige id spezifizierten Videos (z.B. „lecture_8“);
#!/video/:id/:time – Aufruf einer bestimmten Wiedergabeposition time in einem Video (id) im Sinne der Adressierung von Medienfragmenten [Mannens et al. 2012] (siehe TEMPORAL BOOKMARKS und MEDIA FRAGMENTS);
#!/tags/:tag – Übersicht aller Videos mit den angegebenen tags (z.B. „tag1,tag2,tag3“)
– Innerhalb des Widgets vi2.temporal-tags;
```

**R7**

`#!/search/:terms` – Übersicht der Suchergebnisse für die Verknüpfung der angegebenen Suchbegriffe (z.B. „Begriff1,Begriff2,Begriff3“);

`#!/playlist` – Übersicht der in der Playlist enthaltenen Videos.

Die Übersicht an Videos enthält je nach verwendetem HTML-Template Metadaten wie Titel, Autor, Erstellungsdatum, Kurzzusammenfassung und eine Videovorschau in Gestalt von animierten Einzelbildern (siehe VISUAL SUMMARY).

### Suche

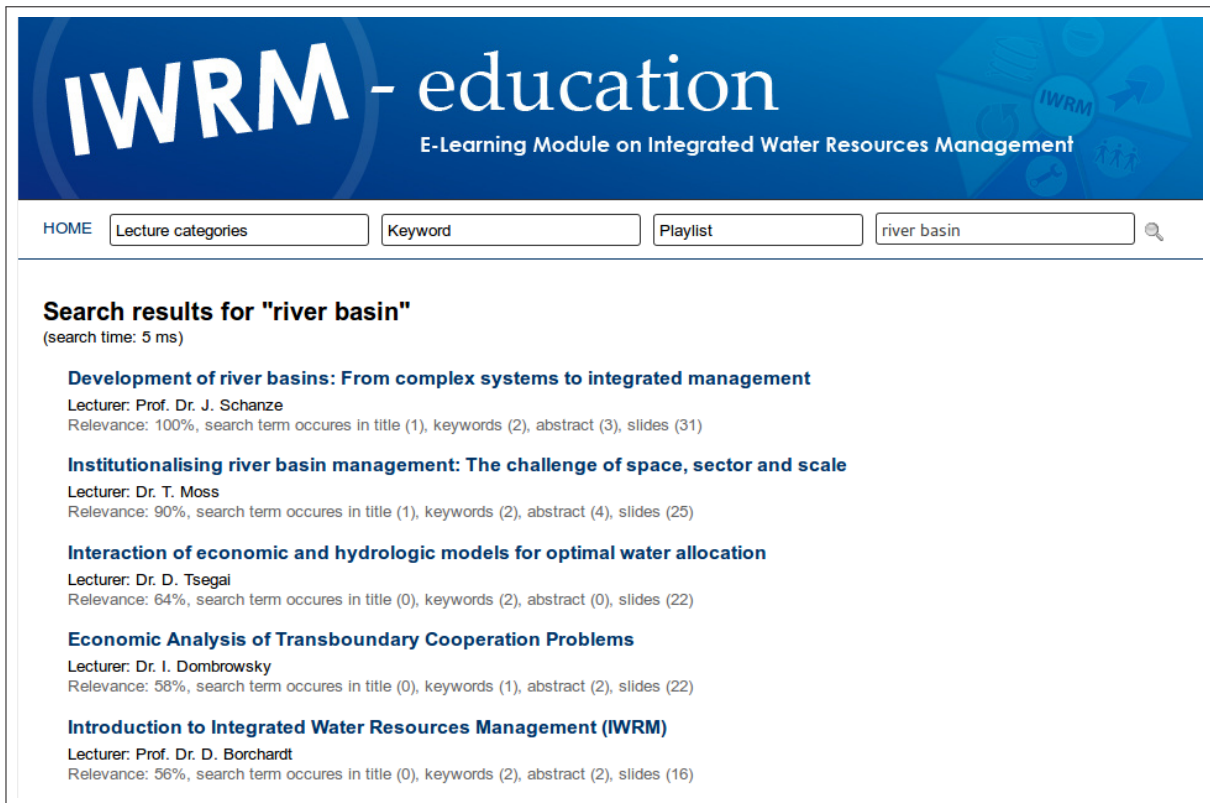
Die Suche nach videografischen Inhalten basiert auf den verfügbaren und individuell gewichteten Informationsquellen:

- R6**
- Metadaten der Videoressourcen, wie z.B. Titel, Autor, Schlagworte / tags oder Kurzbeschreibung;
  - Videoannotationen, wie z.B. Hyperlinks, Kommentare, Kapitelmarken oder zeitgenaue tags;
  - Synchronisierte Medien, wie z.B. per Texterkennung (OCR) extrahierte Wörter von Präsentationsfolien oder Webseiten.

Auf Grundlage dieser Informationen lässt sich eine Kollektion von Videos einschließlich der darin enthaltenen Inhalte im Sinne einer Volltextsuche durchsuchen. Der Suchbegriff kann aus einem oder mehreren Wörtern bestehen. Dabei wird für jedes Video die Auftrittshäufigkeit des Suchbegriffs je vorhandener Informationsquelle ermittelt (z.B. Titel: 1, tags: 2, Folien: 10, Hyperlinks: 2). Falls eine Quelle nicht vorhanden sein sollte, so wird sie ignoriert. Anschließend werden die Quellen gewichtet und aufsummiert und hinsichtlich der der Ergebnisse aller Videos normalisiert, sodass sich die Suchergebnisse nach Relevanz sortiert lassen. Die Gewichtung bildet einen Parameter des Widgets und ist somit anpassbar. Falls sich der Suchbegriff aus mehreren Wörtern zusammensetzt, werden die Wörter einzeln als Suchbegriff verarbeitet und mit den Ergebnissen der übrigen Suchbegriffe zusammengefasst. Eine Suche nach miteinander zusammenhängenden Begriffen wird mit Hilfe *Regulärer Ausdrücke* ermöglicht. Die Ausgabe der Suchergebnisse erfolgt in einer nach Relevanz sortierten Liste im Video Manager. Neben der Relevanz erscheint auch die Trefferanzahl je Suchkriterium in der Liste (siehe Abb. 6.5).

### Playlist

Im VIDEO MANAGER kann ein Video nicht nur aufgerufen, sondern auch für die spätere Betrachtung in einer so genannten Playlist vorgemerkt werden. Die Playlist bildet eine gesonderte Rubrik innerhalb der Anwendung. Die darin enthaltenen Elemente lassen sich einzeln aufrufen oder per Klick aus der Liste entfernen. Mehrere, benutzerdefinierte Playlists werden gegenwärtig noch nicht unterstützt. Die Elemente werden lokal, d.h. unabhängig vom Webserver in einer *indexdb* im Browser gespeichert. Eine Playlist kann deshalb auch nur solange gespeichert werden, wie der Webbrowser geöffnet ist. Wird lediglich das *Tab* oder Browser-Fenster mit der Anwendung geschlossen, so sind die Daten der Playlist bei einem erneuten Aufruf der Anwendung noch verfügbar.



The screenshot shows the IWRM education website interface. At the top is a blue header with the text 'IWRM - education' and 'E-Learning Module on Integrated Water Resources Management'. Below the header is a navigation bar with links: HOME, Lecture categories, Keyword, Playlist, and a search bar containing 'river basin'. The main content area displays 'Search results for "river basin"' with a search time of 5 ms. It lists five search results, each with a title, lecturer, and relevance information:

- Development of river basins: From complex systems to integrated management**  
Lecturer: Prof. Dr. J. Schanze  
Relevance: 100%, search term occurs in title (1), keywords (2), abstract (3), slides (31)
- Institutionalising river basin management: The challenge of space, sector and scale**  
Lecturer: Dr. T. Moss  
Relevance: 90%, search term occurs in title (1), keywords (2), abstract (4), slides (25)
- Interaction of economic and hydrologic models for optimal water allocation**  
Lecturer: Dr. D. Tsegai  
Relevance: 64%, search term occurs in title (0), keywords (2), abstract (0), slides (22)
- Economic Analysis of Transboundary Cooperation Problems**  
Lecturer: Dr. I. Dombrowsky  
Relevance: 58%, search term occurs in title (0), keywords (1), abstract (2), slides (22)
- Introduction to Integrated Water Resources Management (IWRM)**  
Lecturer: Prof. Dr. D. Borchardt  
Relevance: 56%, search term occurs in title (0), keywords (2), abstract (2), slides (16)

**Abbildung 6.5:** Suche innerhalb eines und in mehreren Videos unter Berücksichtigung verschiedener Informationsquellen am Beispiel von *IWRM education* (vgl. [http://www.iwrm-education.de/#!search:river\\_basin](http://www.iwrm-education.de/#!search:river_basin) (abgerufen am 15.06.2015)).

## Related Videos

Zur Ermittlung ähnlicher oder thematisch verwandter Videos können im Widget folgende Kriterien als Parameter herangezogen werden:

- Autor: Videos vom gleichen Autor;
- tags: Videos mit übereinstimmenden Schlüsselwörtern;
- Kategorie: Videos derselben Kategorie;
- Hyperlinks: Videos, auf die Links verweisen (ausgehende Links) oder deren Links das betreffende Video referenzieren (eingehende Links).

Sofern die Anwendung auch serverseitige Daten von Benutzern erfasst, kann auch das Nutzungsverhalten desselben oder der anderen Nutzer als ein Kriterium zur Ermittlung ähnlicher Videos herangezogen werden. Alle Kriterien lassen sich darüber hinaus gewichtet miteinander kombinieren oder als Abstandsvektor in die Ermittlung einbeziehen. Ferner können Elemente der *Related Videos* Destruktoren darstellen, die zufällig ausgewählt werden und somit bewusst von den Kriterien abweichen. Somit besteht die Möglichkeit Anwender zur Auseinandersetzung mit einem anderen Themengebieten zu bewegen. In der Auflistung 6.1 ist ein Beispielcode dargestellt.

```

1  var relatedVideos = new vi2.RelatedVideos( {
2      resultSelector: ".related-videos",
3      limit: 8, # Anzahl der zu ermittelnden Related Videos
4      criteria:[
5          { criterion: "same-author", weight:0.8 },
6          { criterion: "same-tags", weight:0.6 },
7          { criterion: "same-category", weight:0.6 },
8          { criterion: "incomming-links", weight:0.5 },
9          { criterion: "outgoing-links", weight:0.5 },
10         { criterion: "random-destructor", weight:0.1 }
11     ]
12 } );
13 relatedVideos.init();

```

**Auflistung 6.1:** Um eine Menge von acht ähnlichen Videos hinsichtlich der übereinstimmenden Kriterien (1) Autor, (2) *tags*, (3) Kategorie, (4) eingehende und (5) ausgehende Hyperlinks einschließlich zufälliger Destruktoren zu erhalten, sind folgende Anweisungen erforderlich.

## Sequentielle Medien

- R5** SEQUENTIAL MEDIA ist ein Pattern, welches die sequentielle Verknüpfung von Lernvideos im Sinne von Lernpfaden beschreibt. Der Videoplayer ist grundsätzlich dafür ausgelegt, eine Folge von Videos nacheinander wiederzugeben. Das Widget *vi2.sequential-media* unterstützt dabei sowohl Verkettungen, die sich an Kriterien orientieren, als auch benutzerdefinierte Sequenzen von Videos. Im ersten Fall bestehen Parallelen zu den Related Videos, während der zweite Fall den Playlists ähnelt. Zusätzlich zur formalen Verkettung bietet das Widget Navigationsmöglichkeiten entlang der Kette sowie innerhalb eines Rings der aneinandergereihten Elemente an.

## Betrachtungshistorie

Das Pattern VIEWING HISTORY gewährt Anwendern eine chronologische Übersicht der zuletzt aufgerufenen Videos. Sobald eine Videoressource im Player geladen ist, löst dies den *Event* „stream.loaded“ aus. Das Widget *vi2.viewing-history* behandelt den *Event*, indem es den Zeitpunkt des Ereignisses sowie den Bezeichner der Ressource speichert. Bei Bedarf kann der Anwender eine HTTP-Route mit dem Fragmentbezeichner *#!/viewing-history* aufrufen und erhält daraufhin eine Liste der zuletzt aufgerufenen Videos. Als Parameter des Widgets können neben den Videos auch andere Seitenaufrufe verzeichnet werden. Auch für die Anzahl der ausgegebenen Einträge der Historie ist ein Parameter vorgesehen.

### 6.1.7 Autorenwerkzeuge

- Für die Erstellung, Bearbeitung und Verwaltung von interaktiven Videos bietet *VI-TWO* drei Ansätze. Der erste Ansatz basiert auf Auszeichnungen – so genannter *markup*. Auszeichnungen können als HTML-Code (vgl. Auflistung 6.2) oder *Wiki Markup* (vgl. Auflistung 6.3) hinterlegt
- R12**

sein (vgl. auch Seidel [2011a] und Seidel [2012b]). Im Anhang A.18 (S. 344) wird das Regelwerk genauer erläutert.

```

1 <vi2-video      data-src="http://www.videos.com/clip.webm">
2 <vi2-chapter   data-start=9080
3               data-content="Summary"/>
4 <vi2-comment   data-start=34.4
5               data-author="John"
6               data-date="2015-06-28, 12:37"
7               data-content="I agree on that."/>
8 <vi2-link      data-start=435
9               data-duration=20
10              data-target="#!/video/123/4556"
11              data-target-start=30
12              data-target-duration=180 />
13 <\vi2-video>

```

**Auflistung 6.2:** Die als *Custom Elements* in das HTML5-Dokument eingeführten DOM-Elemente für ein Video, eine Kapitelmarke sowie einen Kommentar und einen Link mit den jeweiligen Data-Attributen. Das vi2-video-Element kann dabei mehrere andere Elemente enthalten, die eine Annotation darstellen.

```

1 <hypervideo>
2   [[ Video: http://example.com/video.webm myVideo | 200px ]]
3   [[ Main Page | Startseite ] #10 | 20 | 26% | 26% ]
4   [[ http://www.video-wiki-example.org/ Click me ] #80 | 120 | 50% | 50% ]
5   [[ Video: demo.ogv anotherVideo | 200px ]]
6   [[ Some Page | see here ] #10 | 20 | 26% | 26% ]
7 </hypervideo>

```

**Auflistung 6.3:** Beispiel der um sieben Regeln erweiterten *MediaWiki markup specification* [MediaWiki.org 2015].

Der zweite Ansatz besteht in der direkten Bearbeitung während der Wiedergabe im Player. Um dem Video eine Annotation hinzuzufügen, genügt es eine Schaltfläche zu aktivieren. Daraufhin pausiert das Video und ein Eingabedialog erscheint. Die Anwenderin kann die dortigen Felder ohne Zeitdruck ausfüllen. Erst nach dem Speichern oder Schließen des Dialogs, setzt die Wiedergabe fort. In Abb. 6.6 ist ein exemplarischer Eingabedialog dargestellt.

Als dritten Ansatz bieten mehrspurige Zeitleisten umfassendere Möglichkeiten, um verschiedene Annotationen präzise zeitlich und räumlich zu positionieren und miteinander in Beziehung zu bringen. Die damit verbundene Vorgehensweise wurde bei der Synchronisation von Präsentationsfolien von 42 Lernvideos von *IWRM education* erfolgreich praktiziert [Seidel 2012a]. Ein Screenshot des dafür angepassten Autorensystems ist in Abb. 6.7 zu sehen.

Die Frage, welcher dieser drei Ansätze gegenüber den anderen zu bevorzugen sei, ist vom Anwendungskontext abhängig. In Abhängigkeit der Anzahl an Annotationen skalieren die Ansätze bei der Darstellung von Annotationen sehr unterschiedlich. Mehrspurige Zeitleisten können in Verbindung mit einer Zoom-Funktion eine große Anzahl von Elementen auf den einzelnen Spuren abbilden. Bei einer direkten Manipulation steht jedoch nur eine Zeitleiste mit begrenzter





**Abbildung 6.6:** Eingabedialog in Vi-Wiki zur Definition eines zeitlich und räumlich im Video platzierten Hyperlinks.

horizontaler Ausdehnung zur Verfügung, weswegen dieser Ansatz nur für eine überschaubare Anzahl von Annotationen zu empfehlen ist (z.B. auf Kapitelmarken limitieren). In einem Markup-Editor spiegelt sich eine hohe Anzahl an Annotationen in einer ebenso hohen Zeilenzahl wieder, die für den Anwender beherrschbar ist.

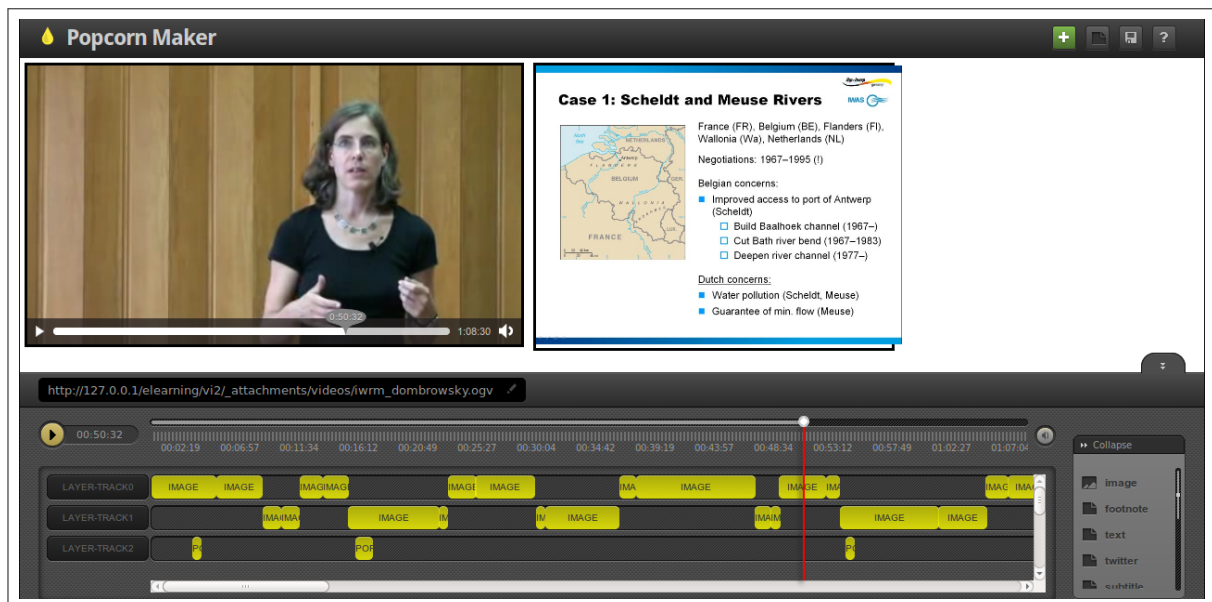
Geht es hingegen darum, sehr viele Annotationen zu bearbeiten, schneidet der Markup-Editor in Verbindung mit gewöhnlichen Editor-Funktionen (z.B. Suchen, Ersetzen, Löschen, Kopieren, Einfügen) besser ab, als die (derzeit verfügbaren) Zeitleisten-Werkzeuge. Mittels der Variante des *Direct Authorings* lassen sich einzelne Annotationen zwar sehr einfach an ihrer zeitlichen Position editieren, jedoch ist die Wiederholung eines solchen Schrittes mit einem hohen Zeitaufwand verbunden.

Als dritte Kontextvariable ist die Bedienbarkeit eine wichtige Größe, wenn es darum geht ein Autorenwerkzeug für weitgehend unbedarfte Anwender zur Verfügung zu stellen. Markup-Repräsentationen scheiden hierbei aus, da Anwender zunächst die Syntax verstehen und erlernen müssen. Auch die Zeitleistenwerkzeuge erscheinen trotz ihrer hohen Verbreitung in einer Vielzahl von Anwendungen (z.B. Videoschnitte, Audibearbeitung) als sehr komplex und wenig übersichtlich.

**Tabelle 6.3:** Vergleichende Einschätzung der drei Ansätze für die Erstellung, Bearbeitung und Verwaltung von Videoannotationen hinsichtlich der Kriterien einer skalierbaren Darstellung, effizienten Bearbeitung und einfachen Bedienbarkeit

	Markup	Direct Authoring	Zeitleiste
Skalierbare Darstellung	hoch	gering	hoch
Effiziente Bearbeitung	hoch	mittel	mittel
Einfache Bedienbarkeit	gering	hoch	mittel

Aus der Gegenüberstellung der drei Ansätze zur Bearbeitung von Videoannotationen in Tab. 6.3 wird deutlich, dass eine Kombination der Ansätze innerhalb einer Anwendung den Bedarf verschiedener Benutzergruppen am ehesten gerecht wird. Während Lernende vornehmlich mit einer überschaubaren Anzahl an Annotationen konfrontiert sind und deshalb *Direct Autho-*



**Abbildung 6.7:** Zur Annotation von Präsentationsfolien und Hyperlinks wurde in IWRM education der Mozilla Popcorn Maker integriert. Auf den oberen beiden Tracks der Zeitleiste repräsentiert die Ausdehnung der einzelnen gelben Blöcke jeweils die Position und Anzeigedauer einzelner Präsentationsfolien. In der unteren Zeitleiste wurden Hyperlinks platziert.

ring-Werkzeuge ausreichen, erfordert die Erstellung und Verwaltung einer Vielzahl von Annotationen leistungsfähiger Werkzeuge, die dem Zeitleisten-Paradigma folgen oder die Markup-Repräsentation des Hypervideos zugänglich machen. In den bisher entwickelten Anwendungen konnte die Bearbeitung mit Hilfe der *Markup* und *Direct Authoring* auch durch mehrere Anwender gleichzeitig erfolgen.

R11

## VI-LAB: CSCL-System für videobasierte CSCL-Scripts

Zur Beantwortung der Frage, wie sich CSCL-Scripts innerhalb einer Videolernumgebung modellieren lassen (Forschungsfrage FF6) wurden in Abschnitt 5.2 (S. 184) bereits ausführliche Angaben über die dafür erforderlichen Komponenten und Mechanismen gemacht. In diesem Teilabschnitt wird VI-LAB als ein CSCL-System vorgestellt, mit der sich videobasierte Scripts modellieren und ausführen lassen. Dabei werden sowohl die Spezifikationen und konzipierten formalen Repräsentationen videobasierter CSCL-Scripts als auch die in Kapitel 4 (S. 91) behandelten Interaction Design Patterns berücksichtigt.

Der Name dieser Lernumgebung setzt sich aus „VI“ für Video und „LAB“ für Labor zusammen. Im Sinne eines Labors soll die Anwendung nicht nur Lehrende in die Lage versetzen, eigenständig kollaborative Lernprozesse anhand von Videos zu strukturieren und in der Lehre einzusetzen, sondern gleichzeitig Wege aufzeigen, neue Variationen von Scripts bzw. neue Scripts zu erproben und zu untersuchen.

### 6.2.1 Anforderungen

Die funktionalen Anforderungen an *VI-LAB* ergeben sich unmittelbar aus der Spezifikation videobasierter CSCL-Scripts. Die Script-Komponenten implizieren Funktionen zur Verwaltung von Scripts, Teilnehmern, Gruppen / Formationen, Lernressourcen, Aufgaben und Aktivitäten sowie den Features des Videoplayers. Für jede der Komponenten bedarf es Übersichten sowie generischer Funktionen, um Komponenten anzulegen, zu aktualisieren oder zu löschen. Auch die Script-Mechanismen zur Gruppenbildung, Sequenzierung von Aufgaben in Phasen, die Verknüpfung von Aufgaben und Ressourcen und die Verteilung der Aufgaben auf Individuen, Gruppen oder alle Nutzer sind erforderlich. In Tab. 6.4 sind alle Anforderungen an *VI-LAB* zusammenfassend dargestellt. Die Nummerierung der Anforderungen (RS1...39) wird im Verlauf der folgenden Seiten diejenigen Textabschnitte kennzeichnen, in denen die Umsetzung der jeweiligen Anforderungen beschrieben ist.

**Tabelle 6.4:** Übersicht der Anforderungen (RS) an *VI-LAB*

#### Script

RS1	CRUD-Operationen zum Anlegen, Anzeigen, Aktualisieren und Entfernen von Scripts, Script-Instanzen und Script-Sessions
RS2	Trennung von Inhalt und Script Dillenbourg und Hong [2008]
RS3	Script-Instanzen: Verknüpfung von Scripts und Lerninhalten (Lernressourcen, Aufgaben, Rollen) und Konfigurationen (Features / konfigurierte <i>VI-TWO</i> -Widgets) Dillenbourg und Hong [2008]
RS4	Script-Session: Durchlauf einer Script-Instanz mit Teilnehmern, Gruppenkonstellationen und unter Angabe des zeitlichen Rahmens Dillenbourg und Hong [2008]
RS5	Verknüpfung mit Teilnehmern, Gruppen, Lernressourcen, Features, Aktivitäten
RS6	Definition von zeitlich begrenzten Phasen
RS7	Terminierte Phasenübergänge durch Angabe eines Zeitpunkts oder Zeitintervalls
RS8	Konditionale Phasenübergänge durch Angabe von Kriterien, die für den Übergang erfüllt sein müssen
RS9	Verzweigte oder adaptive Phasenübergänge in Abhängigkeit der Erfüllung eines Kriteriums
RS10	Freie Phasenübergänge ohne zeitliche oder konditionale Determinierung
RS11	Wiederholung einer Phase mit nur geringen Abweichungen
RS12	Traversion mehrerer Lernressourcen bei gleichbleibender Aufgabenstellung
RS13	Rotation von Rollen beim Phasenwechsel
RS14	Fading von Features, Lernressourcen oder anderen Einschränkungen
RS15	Änderung der zeitlichen Ordnung

(Fortsetzung auf der nächsten Seite)

(Tabelle 6.4 – Fortsetzung von der vorherigen Seite)

**User (Teilnehmer)**

---

- RS16 CRUD-Operationen: Anlegen, Anzeigen, Aktualisieren und Entfernen von Teilnehmern
- RS17 Unterscheidung basaler Nutzerrollen für die Systemadministration
- RS18 Zuordnung von Kriterien, wie z.B. Geschlecht, Studiengang, kultureller Hintergrund
- RS19 Explizite Rollenzuweisung für Teilnehmer

**Gruppe**

---

- RS20 CRUD-Operationen: Anlegen, Anzeigen, Aktualisieren und Entfernen von Gruppen und Gruppenformationen
- RS21 Berücksichtigung der „social planes“ (individuelle Ebene, Gruppen- und Kursebene) Dillenbourg und Hong [2008]
- RS22 Gruppenbildung: Aufteilung in  $n$  Gruppen, Bildung von Gruppen mit maximal  $n$  Personen
- RS23 Kriterien der Gruppenbildung: sequenziell, zufällig, kriteriums basiert (unter Berücksichtigung homogener oder heterogener Kriterien)
- RS24 Berücksichtigung mehrerer Kriterien bei der Gruppenbildung
- RS25 Berücksichtigung externer und interner Kriterien bei der Gruppenbildung
- RS26 Aufteilen und Zusammenlegen bestehender Gruppen

**Lernressourcen / Videos**

---

- RS27 CRUD-Operationen: Anlegen, Anzeigen, Aktualisieren und Entfernen von Lernressourcen
- RS28 Unterstützung verschiedener Typen von Lernressourcen, z.B. Videos/Hypervideos, Bilder, Text/Hypertext
- RS29 Chronologische Segmentierung einer Lernressource

**Features der Lernumgebung**

---

- RS30 Aktivierung / Deaktivierung von Features je Script-Phase, Rolle, Lernressource / Video oder für das gesamte Script
- RS31 Abbildung relevanter *VI-TWO* Widget-Parameter als Features

**Aufgaben / Aktivitäten**

---

- RS32 CRUD-Operationen: Anlegen, Anzeigen, Aktualisieren und Entfernen von Aufgaben
- RS33 Verwaltung der Ergebnisse / Lösungen je Aufgabe
- RS34 Feedback-Möglichkeit je Ergebnis / Lösung durch Lernende und Tutoren / Lehrende

(Fortsetzung auf der nächsten Seite)

(Tabelle 6.4 – Fortsetzung von der vorherigen Seite)

RS35 Unterstützung verschiedener Aufgabentypen unter Verwendung passender *VI-TWO* Widgets

RS36 Unterstützung ausgewählter *thinkLets*

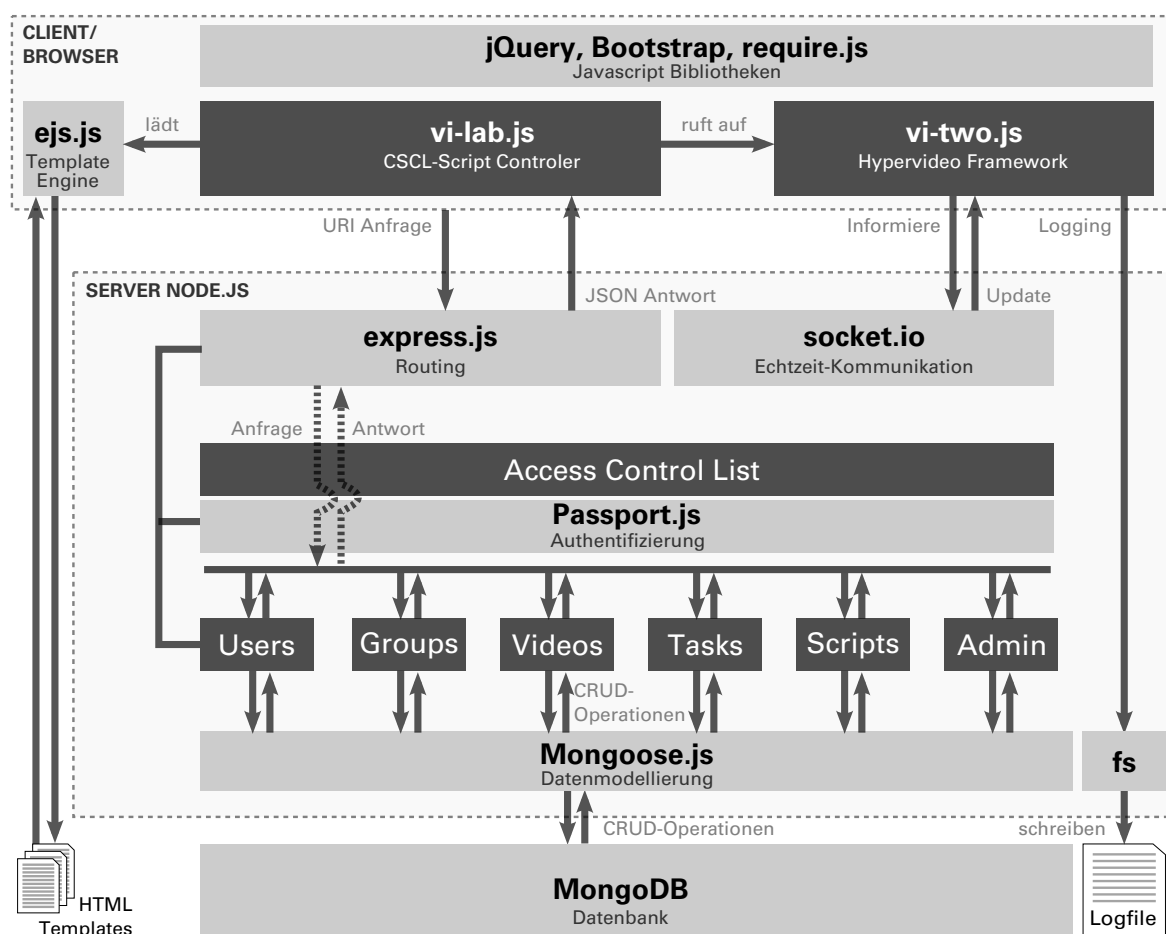
RS37 Aufgaben sind zeitlich im Video verankert bzw. ihnen voran- oder nachgestellt

RS38 Berücksichtigung der „social planes“ in der Verteilung von Aufgaben innerhalb und zwischen Gruppen.

RS39 Gleichzeitige Bearbeitung von Aufgaben in Echtzeit

## 6.2.2 Architektur von VI-LAB

*VI-LAB* baut auf der Laufzeitumgebung *Node.js* und einem damit realisierten Webserver auf. *Node.js* basiert auf *Google V8 JavaScript Engine* und ermöglicht eine ereignisbasierte Programmierung skalierbarer Echtzeit-Anwendungen im Web. Befördert wird dies insbesondere durch die nicht-blockierende, asynchrone Verarbeitung von Anweisungen. Die Architektur von *VI-LAB* ist in Abb. 6.8 dargestellt und wird in den nachfolgenden Teilabschnitten anhand ausgewählter Aspekte näher erläutert.



**Abbildung 6.8:** Client-Server-Architektur von *VI-LAB* unter Angabe der verwendeten Frameworks bzw. Module (hellgrau) sowie der eigens entwickelten Komponenten (dunkelgrau).

## Datenhaltung mit *MongoDB* und *Mongoose*

Zur Datenhaltung wird *MongoDB* als dokumentenbasierte Datenbank eingesetzt. Ein Dokument besteht dabei aus beliebigen JSON- oder BSON-Daten. Dokumente inklusive ihrer Felder, Datentypen, Feldgrößen und Beziehungen lassen sich somit auf einfache Weise direkt im Code von *Node.js* modellieren, ohne vorher unbedingt ein Datenbank-Schemata definieren zu müssen (schemafrei). Dokumente können somit verschieden aufgebaut sein, wobei sie stets durch eine eindeutige ID identifizierbar sind. Als Abstraktionsebene zur leichteren Validierung, Strukturierung und Abbildung von Datenobjekten bildet *Mongoose* die Schnittstelle zum nativen *MongoDB*-Treiber in *Node.js*.

## REST API und *Express.js* Routing

Die Kommunikation zwischen Client und Server basiert in *VI-LAB* fast vollständig auf einer *REST API*. Jede, vom Client angefragte Web-Adresse (URL) repräsentiert dabei eine bestimmte Struktur von Informationen. Alle Informationen sind somit adressierbar.

```

1 # HTTP GET
2 /videos                                # Video Manager
3 /videos/view/:id                       # Videoplayer / VI-TWO
4 /admin/videos                          # Liste aller Video mit Metadaten
5 /admin/videos/new                      # Formular fuer ein neues Video
6 /admin/videos/metadata/edit/:id        # Formular der Metadaten eines Videos
7 /admin/videos/annotate/edit/:id       # Formular der Annotationen eines Videos
8 /json/videos                           # JSON: verfuegbare Videoinstanzen
9 /json/videos/:id                       # JSON: ein Video
10 /json/admin/video-instances            # JSON: alle Videoinstanzen
11 /json/admin/video-files                # JSON: alle Videodateien
12
13 # HTTP POST
14 /videos/create                         # Anlegen eines neuen Videos
15 /videos/update/:id                     # Aktualisierung eines Videos
16 /videos/destroy/:id                   # Entfernen eines Videos
17 /videos/annotate                       # Speichern einer neuen Annotation

```

**Auflistung 6.4:** Die über *Express.js* definierten HTTP Routen für die Operatoren GET und POST am Beispiel der Script-Komponente Video.

Mit Hilfe des Frameworks *Express.js* sind die HTTP-Routen serverseitig definiert und jeweils mit den entsprechenden Routinen in den einzelnen Modulen der Script-Komponenten verknüpft. In den Modulen werden die Anfragen durch Datenbankabfragen in eine bestimmte Datenstruktur überführt und als JSON- oder HTML-Dokument ausgeliefert. JSON-Dokumenten werden vom Client weiter verarbeitet und zur Anzeige gebracht, während HTML-Dokumente mit Hilfe der Template-Engine *ejs.js* gerendert und dann im Web-Browser ausgegeben werden. JSON-Dokumente enthalten beispielsweise Statusinformationen eines Nutzer oder die Script-Konfiguration für die Auslieferung eines Videos inklusive der vorgesehenen Annotationsfunktionen. Ein wesentlicher Bestandteil dieser REST-Architektur sind die in den Antworten des Servers enthaltenen Hypermedia-Links zu anderen REST-Schnittstellen des Systems. Dieses,

als HATEOAS bezeichnete Prinzip, erleichtert die Referenzierung von Ressourcen und CRUD-Operation auf der Seite des Clients. URLs müssen somit nicht innerhalb des Clients konstruiert und für jedes Template oder jede AJAX-Anfrage angepasst werden. In der Auflistung 6.4 sind exemplarisch alle HTTP-Routen der Script-Komponente *Video* aufgelistet.

## Kommunikation und Annotation in Echtzeit

Die Anforderung, neben asynchronen auch synchrone Arbeitsweisen zu unterstützen, ergab sich aus den anvisierten Gruppenarbeiten. Erfahrungen mit dem ersten Prototyp hatten gezeigt, dass sowohl für die Kommunikation innerhalb der Gruppe, als auch für die Annotationen des Videos stets der aktuellste Bearbeitungsstand vorliegen muss. Ist dies nicht der Fall, kann es zu Missverständnissen in Diskussionen oder zu Dopplungen bei Annotationen führen. Die Mindestanforderung besteht darin, sehen zu können, wer gerade im System eingeloggt ist. Diese Information ist für den Anwender wichtig, um von den Aktivitäten anderer Nutzer Kenntnis nehmen zu können und gegebenenfalls in Kontakt zu treten oder eine Aufgabe gleichzeitig zu bearbeiten. Zudem ist es erforderlich, bestimmte Benutzereingaben unmittelbar den übrigen Gruppenmitgliedern sichtbar zu machen, damit es zu keinen Dopplungen kommt. Dies ist zum Beispiel bei der Annotation von Kapitelmarken der Fall.

**RS39** Mit dem Framework *socket.io* konnte diese Anforderung in *VI-LAB* folgendermaßen erfüllt werden: Nach dem Login eines Nutzer versendet *socket.io* eine Broadcast-Mitteilung an alle Clients der Nutzer, die derselben Benutzergruppe angehören. *Socket.io* organisiert diese Nutzer in sogenannten *rooms* oder Namensräumen, denen ein Nutzer nach dem Login beitrifft (*join*). Die Broadcast-Mitteilung enthält die Nutzer-ID, sodass im Client-System ein Icon für den neu hinzugekommenen Nutzer oben rechts im Browser-Fenster angezeigt werden kann. In der Datenbank wird der Online-Status des Nutzer zudem serverseitig registriert. Alle Nutzer einer Gruppe, die gleichzeitig online sind, können dann z.B. miteinander chatten. Der Chat funktioniert ebenfalls auf Basis der *rooms* in *socket.io*. Annotiert ein Nutzer etwas im Video, speichert der Server dies in der Datenbank. Gleichzeitig werden alle Clients der Nutzer, die aktuell das gleiche Video aufgerufen haben und sich deshalb im gleichen *room* befinden, über ein Update der Annotationen informiert. Daraufhin rufen sie automatisch den aktuellen Stand der Annotationen vom Server ab und stellen ihn wieder im Videoplayer dar. Für den Anwender ergeben sich daraus keine merklichen Verzögerungen bei der Datenübertragung und Wiedergabe des Videos. Loggt sich ein Nutzer aus, versendet *socket.io* abermals eine Broadcast-Mitteilung an alle Clients der Gruppenmitglieder, um seinen Online-Status zu ändern.

## Datenverwaltung der Script-Komponenten

Die Datenverwaltung der Script-Komponenten umfasst die CRUD-Operatoren zum Erstellen, Lesen, Aktualisieren und Löschen (*create, read, update, delete*) von MongoDB-Dokumenten. Anwendern werden entsprechende Eingabemasken und Ansichten für Videodateien, Videoinstanzen, Teilnehmer, Gruppen, Scripts, Script-Instanzen sowie alle Formen von Annotationen, Aufgaben und Aufgabenlösungen angeboten. In Abb. 6.9 ist eine beispielhafte Ansicht des Administrationsbereichs zu sehen.

**RS27**

**RS16**

**RS20**

**RS1**

**RS27**

**RS32, RS33**



Zur Datenverwaltung sind ausschließlich Nutzer berechtigt, die der Rolle „Editor“ zugeordnet sind. Darüber hinaus gibt es die Nutzerrolle „Teilnehmer“, welche lediglich scriptgesteuerte Lernphasen durchlaufen können. **RS17**

ID	Titel	File	Annotations	Last Update	
1	Video 1: Einführung VCL und Blended Learning	<a href="#">link</a>	chapter marks (1) slides (28)	4 minutes ago	<a href="#">▶ view</a> <a href="#">✎ edit annotations</a> <a href="#">✎ edit meta data</a> <a href="#">✕ remove video instance</a>
10	Video 2: Kommunikation und Kommunikationsmedien im virtuellen Raum	<a href="#">link</a>	slides (33)	22 days ago	<a href="#">▶ view</a> <a href="#">✎ edit annotations</a> <a href="#">✎ edit meta data</a> <a href="#">✕ remove video instance</a>
11	Video 2: Kommunikation und Kommunikationsmedien im virtuellen Raum	<a href="#">link</a>	slides (33)	22 days ago	<a href="#">▶ view</a> <a href="#">✎ edit annotations</a> <a href="#">✎ edit meta data</a> <a href="#">✕ remove video instance</a>
12	Video 2: Kommunikation und Kommunikationsmedien im virtuellen Raum	<a href="#">link</a>	slides (33)	22 days ago	<a href="#">▶ view</a> <a href="#">✎ edit annotations</a> <a href="#">✎ edit meta data</a> <a href="#">✕ remove video instance</a>

**Abbildung 6.9:** Übersicht der Videoinstanzen im Administrationsbereich von VI-LAB. Dargestellt ist der Titel der in der Instanz genutzten Videodatei sowie die Quantität der Annotationen je Typ.

### 6.2.3 Modellierung videobasierter CSCL-Scripts

Die Modellierung eines Scripts mit einer Komplexität auf Level 3 erfordert fünf Schritte. Davon ausgenommen die vorbereitende Konfigurationen im System, um etwa Videos und Nutzer einzupflegen.

**Vorbereitung** Die Ausgangssituation für die Modellierung eines Scripts besteht aus einer Menge verfügbarer Videos. Diese werden entweder durch die Angabe einer URL referenziert oder auf den Server hochgeladen. In beiden Fällen sind für die Videodateien Metadaten (*Dublin Core*) wie Titel, Autor, Beschreibung anzugeben. VI-LAB erzeugt durch eine Schnittstelle zu *libav* automatisch Vorschaubilder (*avconv*) und extrahiert technische Metadaten (*avprobe*). Im Hintergrund erfolgt eine Konvertierung in die für die Auslieferung in gängigen Web-Browsern erforderlichen Video-Codecs.

Des Weiteren bedarf es einer Menge an registrierten (Kurs-)Teilnehmern, für die neben den typischen persönlichen Angaben auch weiterführende Kriterien (z.B. Geschlecht, Studiengang) hinterlegt sein können. Die Eingabe kann händisch per Formular oder durch den Import einer entsprechenden Tabelle im CSV-Format erfolgen. Aus der Menge aller oder ausgewählter Teilnehmer müssen anschließend Gruppen gebildet werden. Für die Gruppenbildung sind mehrere Verfahren implementiert, um eine gewünschte Anzahl von Gruppen oder eine bestimmte Gruppengröße zufällig, sequenziell oder nach bestimmten Kriterien der Teilnehmer zu formieren. **RS18**

**RS22**  
**RS23, RS24**

Eine Gruppenbildung unter Berücksichtigung externer Kriterien (RS25) sowie die kriterienbasierte Aufteilung von Gruppen (RS26) wurde jedoch noch nicht umgesetzt.

Im Übergang von einer Phase zu nächsten müssen die bestehenden Gruppen nicht unbedingt beibehalten werden. Aus diesem Grund ist es auch möglich bestehende Gruppen aufzuteilen

**RS26** oder mehrere zusammenzulegen. Eine Zusammenlegung zweier Gruppen kann dabei auch die Verschmelzung zweier Videoinstanzen zur Folge haben, wenn etwa zwei Gruppen nach der Annotation der gleichen Videodatei ihre Ergebnisse gemeinsam diskutieren sollen.

**Schritt 1: Script-Template auswählen oder erstellen** Ausgehend von der von Dillenbourg und Hong [2008] vorgeschlagenen Unterscheidung von Script, Script-Instanz und Script-Session muss bei der Modellierung eines Scripts zunächst ein Template ausgewählt werden. Optional lassen sich Templates auch neu erzeugen oder duplizieren und, wie in Abb. 6.10 zu sehen, bearbeiten. Ein Template enthält dabei basale Angaben zu den einzelnen Phasen. Folgende Angaben sind je Phase zu machen:

- Titel / Bezeichnung;
- Aufgabenstellung;
- Zusatzmaterialien („Supplements“), wie z.B. Texte, Links oder Bilder;
- verfügbare Features und *thinkLets* in Form von Widgets, die in ihrer anzugebenden Konfiguration zunächst für die gesamte Phase gelten;
- zu nutzende Videodateien.

Neben den auf das Video sowie konvergente Gruppenaktivitäten bezogenen Widgets von *VI-TWO*, wurden ausgewählte *thinkLets* (*BucketShuffel* und *StrawPoll*) ebenfalls als Widget implementiert und zur Auswahl gestellt. Auf diese Weise kann eine Gruppe beispielsweise dazu aufgefordert werden, die Beiträge der Gruppenmitglieder intern auf einer Skala von 1 bis 5 zu bewerten.

**RS36**

**Schritt 2: Script-Session** Um eine neue Script-Session anzulegen, muss ein bestehendes Template zunächst instantiiert werden. Die Angaben des Templates werden übernommen und müssen angepasst werden. Zwingend erforderlich ist die Auswahl einer Gruppenformation.

**RS6**

**RS7**

**RS10**

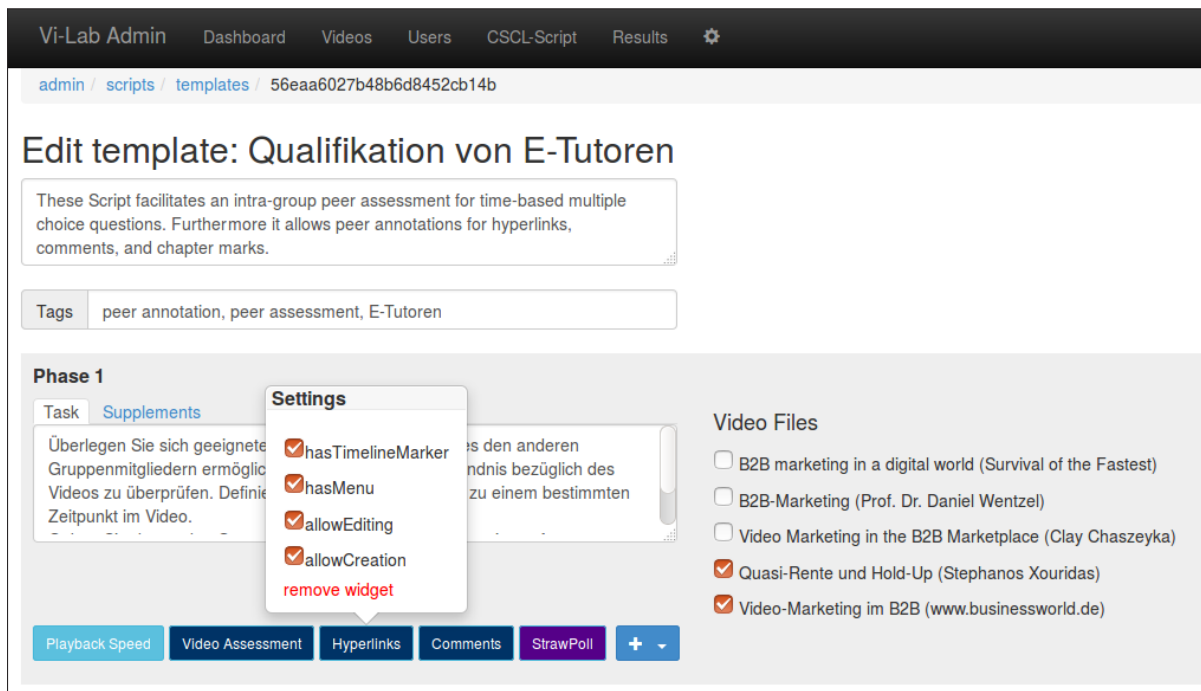
Auch die Angaben zur zeitlichen Ausdehnung bzw. dem genauen Beginn und Ende sind empfehlenswert, wenn auch optional. Konditionale oder verzweigte Phasenübergänge (RS8, RS9) wurden noch nicht realisiert.

Je nach Komplexitätslevel werden aus den Videodateien mit ihren hinterlegten Metadaten Instanzen erzeugt, die unabhängig von den übrigen Instanzen desselben Videos bearbeitet werden können. In einem Script auf dem Komplexitätslevel 2 wird für jede Gruppe je eine Instanz der ausgewählten Videodateien automatisch erzeugt (siehe Abb. 6.11). Auf Komplexitätsstufe 3 entscheidet der Anwender in den folgenden beiden Arbeitsschritten, welche Instanzen für welche Gruppen erzeugt werden sollen.

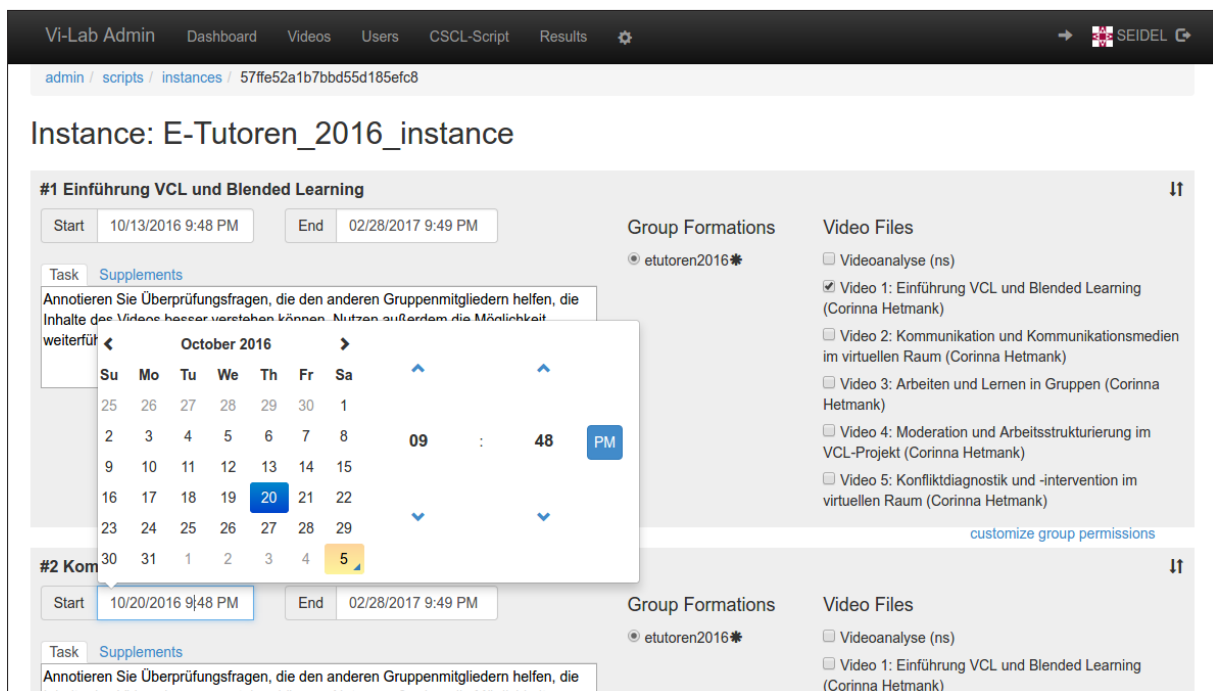
**Schritt 3: Zugänglichkeit** In einer gesonderten Matrix wird in einem Level-3 Script für je-

**RS5**

de Gruppe festgelegt, auf welche Videoinstanzen sie zugreifen darf. Auf diese Weise können je Script-Phase sowohl individuelle, kollektive und kollaborative Lernaktivitäten in Bezug auf ein oder mehrere Videos modelliert werden. Individuelle Aktivitäten erfordern Gruppen mit ge-



**Abbildung 6.10:** GUI für die Erstellung und Bearbeitung eines Script-Templates. Im dargestellten Beispiel sind ein Player-Widget („Playback Speed“), drei Annotations-Widgets („Comments, Hyperlinks, Video Assessment“) sowie das *thinkLet* „StrawPoll“ für Phase 1 ausgewählt.



**Abbildung 6.11:** GUI für die Bearbeitung von Script-Instanzen im Administrationsbereich von *VI-LAB*. Neben dem Beginn und Ende sowie der Aufgabenstellung der Script-Phasen können eine Gruppenformation und mehrere Videodateien ausgewählt werden.

nau einem Teilnehmer, denen jeweils eine Videoinstanz zugeordnet ist. Kollektive Aktivitäten entstehen, wenn alle Teilnehmer in einer Gruppe vereint sind, während sie sich für das kollaborative Lernen auf mehrere Gruppen aufteilen, die jeweils unterschiedliche Instanzen bearbeiten. Dieser Schritt wird für jede Script-Phase wiederholt.

**Schritt 4: Widget-Parameter** Für jede in Schritt 3 einer Gruppe zugeordnete Videoinstanz werden nun die *VI-TWO-Widgets* ausgewählt, die später den Gruppenmitgliedern zur Verfügung stehen sollen. Je Widget lassen sich spezifische Parameter für die jeweilige Gruppe und die Videoinstanz festlegen. Dadurch kann eine Instanz beispielsweise durch eine Gruppe mit Überprüfungsfragen angereichert werden, während eine andere Gruppe lediglich die Fragen bearbeiten darf. Festlegungen für Widgets und deren Einstellungen lassen sich darüber hinaus auch zeilenweise für alle Gruppen sowie spaltenweise für alle Instanzen fixieren. Falls alle Instanzen für alle Gruppen mit den gleichen Features erscheinen sollen, so kann dies für die gesamte Matrix angegeben werden.

Neben den auf das Video sowie konvergente Gruppenaktivitäten bezogenen Widgets, wurden ausgewählte *thinkLets* (*BucketShuffel* und *StrawPoll*) implementiert und zur Auswahl gestellt. Auf diese Weise kann eine Gruppe beispielsweise dazu aufgefordert werden, die Beiträge der Gruppenmitglieder intern auf einer Skala von 1 bis 5 zu bewerten.

**Schritt 5: Aktivierung** Eine gespeicherte Script-Instanz ist erst dann für die Teilnehmer nutzbar, wenn sie im Adminbereich aktiviert worden ist und der Starttermin (Datum und Uhrzeit) mindestens einer Script-Phase erreicht ist. Eine solche laufende Script-Session kann dann durch den Lehrenden oder einen Tutor im Analytical Dashboard beobachtet werden.

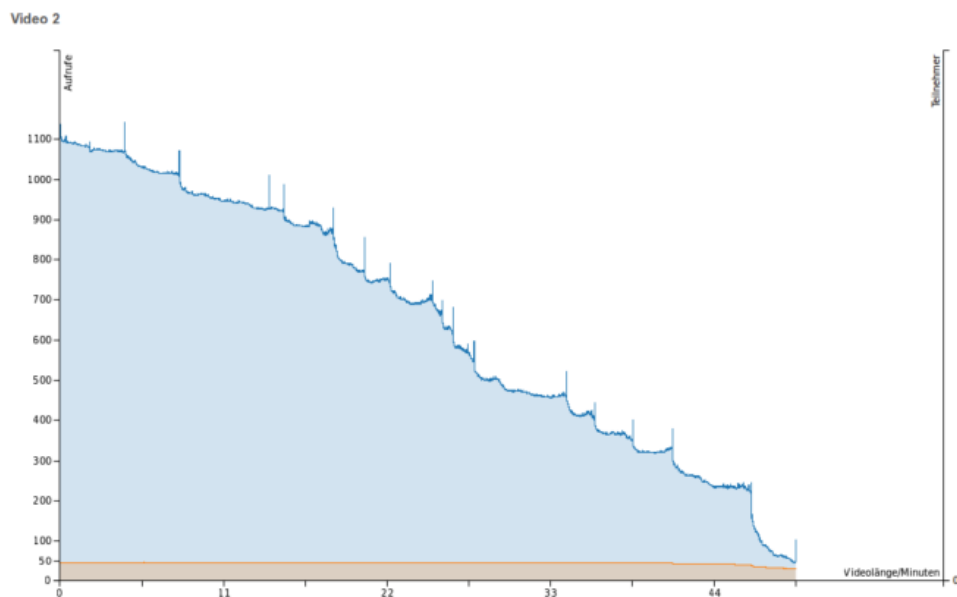
Das eben geschilderte Prozedere kann gewisse Sequenzierungen von Lernaktivitäten in Bezug auf die Zuweisung von Features und Lernressourcen nicht automatisieren. Eine Wiederholung einer Phase mit geringfügiger Abweichung sowie eine Traversalion und Rotation von Rollen und Ressourcen muss, ebenso wie ein Fading, manuell umgesetzt werden (siehe Anforderungen RS11, RS12, RS13 und RS14). Der hier vorgestellte Ansatz substituiert teilnehmerbezogene Zuweisungen von Rollen durch das Konzept von Gruppen, die gegebenenfalls auch aus nur einer Person bestehen können.

#### 6.2.4 Monitoring

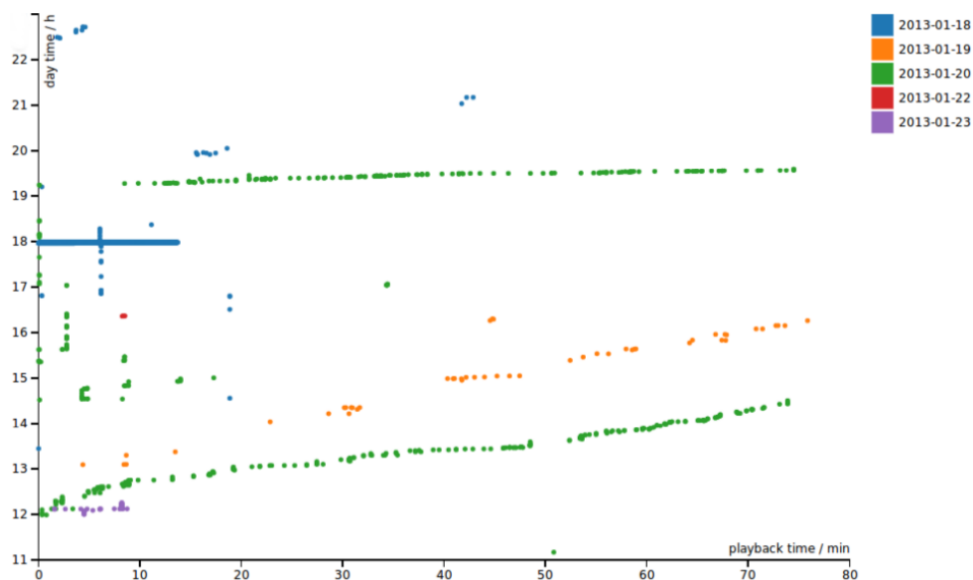
Abgesehen von der wissenschaftlichen Untersuchung des Lernverhaltens, dient das Monitoring dazu, den reibungslosen Ablauf der computergestützten Gruppenarbeit zu gewährleisten, um bei Problemen nötigenfalls intervenieren zu können.

Für die Erhebung des Nutzerverhaltens zeichnet *VI-LAB* zum einen die vom Nutzer ausgelösten Ereignisse wie z.B. Seitenaufrufe, Steuerungsfunktionen im Video oder Annotationen auf. Während der Videowiedergaben erfolgt zum anderen eine Protokollierung der vollständig abgespielten Segmente (je 5 Sekunden) eines Videos. Die aufgezeichneten Daten werden in *MongoDB* abgelegt und mit Hilfe einer speziell entwickelten Analysesoftware (*vi-analytics*) aufbereitet und in Form von Kennzahlen und Datengrafiken ausgegeben.

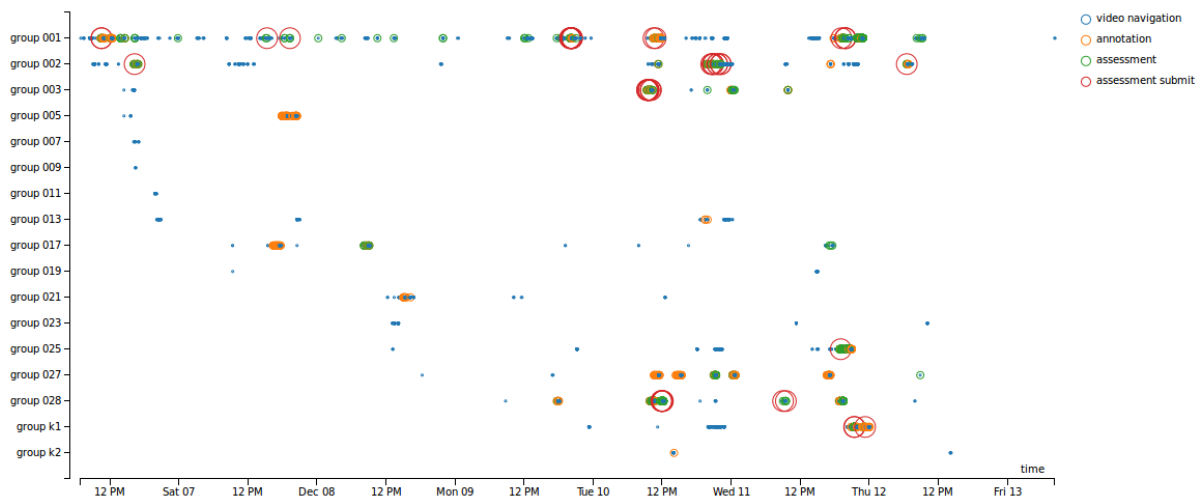
In einem so genannten *Analytical Dashboard* werden die Ausgaben mit Hilfe von *D3.js* und *C3.js* visualisiert. Dabei wurden nutzerbezogene, gruppenbezogene und videobezogene Analysen unterschieden. Ausführliche Erläuterungen zu den hierfür entwickelten Datenvisualisierungen finden sich bei Seidel [2013, 2014a]. In Abb. 6.12 bis 6.14 sind ausgewählte Ansichten dargestellt.



**Abbildung 6.12:** *Absolute Retention Rate* je Video: Die Anzahl der Aufrufe einzelner Videosegmente ist in Abhängigkeit der Abspielposition als blaue Fläche dargestellt. In der unteren, grau hinterlegten Fläche, ist die Anzahl der Teilnehmer abgetragen, die ein Segment des Videos betrachtet haben. Siehe auch Seidel [2014a].



**Abbildung 6.13:** *Rewatching Graph* [Brooks et al. 2013] je Teilnehmer und Video: Auf der Y-Achse ist die Tageszeit der protokollierten Ereignisse der Wiedergabezeit auf der X-Achse gegenübergestellt, während die unterschiedlichen Tage farblich kodiert sind. Siehe auch Seidel [2014a].



**Abbildung 6.14:** Im *CORDTRA*-Diagramm [Hmelo-Silver et al. 2011] sind die Aktivitäten der einzelnen Gruppen im Verlauf der Zeit gegenübergestellt. Das hier dargestellte Beispiel zeigt die Aktivitäten der einzelnen Gruppen im zeitlichen Verlauf anhand folgender Markierungen: Navigation (blaue Punkte), Annotation von Kommentaren / Kapiteln / Tags (orangene Punkte) sowie das Erstellen (grüne Punkte) und Bearbeiten (rote Ringe) von Assessment-Aufgaben.

## Anwendungsbeispiele für VI-TWO und VI-LAB

In diesem Abschnitt werden fünf Software Lösungen kurz vorgestellt, welche die flexiblen technischen Anwendungsmöglichkeiten des Hypervideo Frameworks *VI-TWO* und der CSCL-Umgebung *VI-LAB* verdeutlichen.

### 6.3.1 Vi-Wiki

Vi-Wiki stellt eine Erweiterung (*extension*) für das weit verbreitete Wiki-System *MediaWiki* (u.a. *Wikipedia* basiert darauf) dar. Damit wurde versucht, das Konzept der Verlinkung innerhalb und zwischen Wiki-Artikeln auch auf die in den Artikeln enthaltenen Videos zu übertragen. Im Ergebnis können Links zeitlich und räumlich im Video platziert werden. Die Verweise können sowohl auf Abschnitte innerhalb desselben oder auf einen anderen Artikel oder eine externe Ressource verweisen. Das Besondere an der Implementierung ist die Möglichkeit, Links direkt während der Betrachtung eines Artikels definieren zu können, sodass sie auch im Quellcode des Artikels – der Wiki Markup – erscheinen. Somit können alle inhaltlichen Änderungen im Revision-Log des Artikels erfasst, nachvollzogen und bei Bedarf rückgängig gemacht werden.

Bisheriger Einsatz:

Als Teil einer Geschichtsdokumentation wurden fünf videografierte Zeitzeugeninterviews nebst Texten in ein Wiki integriert und im Sommer 2012 als Ergänzungsangebot den jugendlichen Teilnehmern einer *Summer School* der *Aktion Sühnezeichen Friedensdienste* in Görlitz angeboten.

Quellcode: <https://github.com/nise/vi-wiki> (abgerufen am 22.01.2016).

Weiterführende Informationen: Seidel [2011a, 2012b].

### 6.3.2 IWRM education

Zwischen den Jahren 2011 und 2013 wurde ein E-Learning Modul aus 43 Vorlesungsvideos zum Thema „Integrated Water Resources Management“ (IWRM) entwickelt. Das Projekt koordinierte das Sekretariat des *International Hydrological Programme* (UNESCO) und *Hydrology and Water Resources Programme* (World Meteorological Organization). Am Projekt beteiligten sich bis dato 34 Lehrende von 21 nationalen und internationalen Hochschulen und Forschungseinrichtungen. Die Beteiligten steuerten jeweils ein bis drei Lernvideos im Umfang von 58 bis 158 Minuten bei. Zielgruppe des informellen Lernangebots sind Studierende sowie Entscheidungsträger/innen in Schwellen- und Entwicklungsländern. Der bis dahin erste Ansatz, Lernangebote verschiedener Fachgebiete zum Thema Wasserbewirtschaftung in einer Lernumgebung zu integrieren, wurde durch über 250 zeitgenaue Verweise zwischen den Videos verstärkt. Diese Form einer nicht-linearen Videonavigation wird durch Inhaltsverzeichnisse je Video, *tags* sowie eine Volltextsuche in den zum Video synchronisierten Präsentationsfolien untermauert. Die Anwendung lässt sich sowohl im WWW als auch ohne Internetverbindung von einem portablen Speichermedium in gängigen Web-Browsern aufrufen.

Bisheriger Einsatz (Auswahl):

- IWRM-Workshop an der Universität Kiel (2013).
- Das Modul war in den Jahren 2011, 2012 und 2013 Bestandteil der vom Helmholtzzentrum für Umweltforschung Leipzig durchgeführten *IWRM summer schools*<sup>18</sup>.
- Als ein unterstützendes Lehrmittel im Wahlpflichtmodul „Integriertes Wasserressourcenmanagement (IWRM)“ wird das Modul für die Masterstudiengänge der Fachrichtung Hydrowissenschaften an der Technischen Universität Dresden seit 2013 eingesetzt.
- Im Rahmen des BMBF-Projekts *International Water Research Alliance Saxony* wurde das Modul als ein unterstützendes Lehrmittel in einem IWRM-Kurs an der *National University of Water Management and Nature Resources Use*<sup>19</sup> in Rivne/Ukraine eingesetzt.
- Als *Open Educational Resource* steht das Modul als informelles Online-Lernangebot seit März 2013 unter <http://www.iwrm-education.org/> (abgerufen am 22.01.2016) zur Verfügung.

Quellcode: <https://github.com/nise/vi-iwrm> (abgerufen am 22.01.2016).

Weiterführende Informationen: z.B. Leidel et al. [2012]; Seidel [2011b, 2012a].

### 6.3.3 VI-LAB (Version 1) auf Basis von Wordpress

Als Basis für diese erste Implementierung von *VI-LAB* diente die Mehrbenutzer-Variante der Blog-Software *Wordpress*. Dieses CMS bietet einer Reihe essentieller Funktionen für die Benutzerverwaltung, das Rechteverwaltung und die Pflege von Inhalten. Erweitert wurde *Wordpress* durch ein eigenentwickeltes Plugin, welches den Script-Editor im Backend und den Script-Player im Frontend beinhaltet. Der Script-Editor erlaubt die Definition von Script-Phasen durch die Angabe von Aufgabenstellungen, zeitlichen Begrenzungen sowie Parametern der jeweils

<sup>18</sup>DAAD Summer School on IWRM, 10.-23.09.2012 in Dessau-Roßlau und Berlin; DAAD-Summerschool on Integrated Water Resources Management, 30.9.-11.10.2013 in Magdeburg und Berlin; DAAD-Summerschool on Integrated Water Resources Management, 8.-19.9.2014 in Dessau und Berlin.

<sup>19</sup>Siehe <http://nuwm.edu.ua/en> (abgerufen am 17.06.2015).



zur Verfügung stehenden Player-Widgets von *VI-TWO*. Im Script-Editor lassen sich die Widgets aktivieren bzw. deaktivieren. Auch die Darstellungsform (im Video, neben dem Video, auf dem Folien-Bereich oder auf der Zeitleiste) und der Annotationsmodus konnten angepasst werden. Aufgrund der unzureichenden Medienunterstützung von *Wordpress* wurde das Backend außerdem um Funktionen zum Hochladen von Videos und zur Verwaltung von audiovisuellen Medien ergänzt. Teil dessen ist eine Übersicht über alle, in den Nutzer-Blogs verfügbaren Videos bzw. Beiträge, in denen die Videos enthalten sind. Der Script-Player stellt das Lernvideo gemäß den Einstellungen im Editor bereit und ergänzt es um zeitliche Annotations- und Navigationsmöglichkeiten. Die Bearbeitung der Annotationen (Kapitelmarken, *tags*, Überprüfungsfragen, Kommentare) konnte entweder durch Benutzerdialoge im Player oder im Zeitleisten-Dialog des integrierten *Popcorn Maker* erfolgen.

Bisheriger Einsatz:

- In einer Feldstudie am Internationalen Hochschulinstitut Zittau bearbeiteten im Wintersemester 2012/2013 32 Studierende im Rahmen der Vorlesung „Supply Chain Management“ eines wirtschaftswissenschaftlichen Masterstudiengangs innerhalb von 14 Tagen ein videobasiertes CSCL-Script zur Peer Annotation von zeitgenauen Kapitelmarken und *tags* sowie dem Peer Assessment mittels zeitgenauen Multiple-Choice-Fragen.

Quellcode: <https://github.com/nise/vi-press> (abgerufen am 22.01.2016).

Weiterführende Informationen: Seidel [2013]

### 6.3.4 VI-LAB (Version 2) auf Basis von node.js

In der zweiten Version wurde *Wordpress* (inkl. *MySQL*) im Backend durch *node.js* und *MongoDB* ersetzt. Im Vergleich zur vormals eingesetzten Lernumgebung (Version 1) konnten nun mehrere Personen gleichzeitig Videos annotieren und die Bearbeitungen ihrer Peers unmittelbar verfolgen. Diese Echtzeit-Kommunikation zwischen dem Server und mehreren Clients beruht auf *socket.io*.

Bisheriger Einsatz an der TU Dresden:

- SCM-LAB 2: In einer Feldstudie im Wintersemester 2013/14 mit insgesamt 76 Studierenden zweier Studiengänge unterschiedlicher Hochschulen wurden Aufgaben in Form eines *Peer Assessments* und einer *Peer Annotation* bearbeitet. Die Zusammenarbeit von 40 Studierenden wurden dabei durch ein CSCL-Script gesteuert, während die übrigen 36 ohne eine getaktete Zuteilung von Lernressourcen, Funktionalitäten und Gruppenkonstellationen die gleichen Aufgaben bewältigten.
- Im Rahmen des Projektes *E-TuScript* wurde damit begonnen, den ersten Teil der Lehrveranstaltung „Qualifikation von E-Tutoren“ in eine script-gesteuerte Selbstlernphase zu überführen. Im Zeitraum von fünf Wochen bearbeiteten 55 Studierende jeweils in Kleingruppen wöchentlich ein Video. Die gemeinschaftlich zu bearbeitenden Aufgaben bestanden darin, die Videos durch Kapitelmarken zu annotieren, die Inhalte zu diskutieren und die verankerten Überprüfungsfragen („on demand tasks“ und Lückentextaufgaben) zu lösen. Zusätzlich wurden kurze Fallstudien bearbeitet.

- Im Rahmen der Lehrveranstaltung „Qualifikation von E-Tutoren“ wurde im Wintersemester 2015/16 eine fünfwöchige Videolernphase erneut implementiert. Eine Kohorte von 24 Studierenden, aufgeteilt in sechs Gruppen, annotierte wöchentlich ein neues Video mit insgesamt 33 Verweise auf weiterführende Lernmaterialien, 171 Überprüfungsfragen als Selbsttest innerhalb der Gruppen und 65 Kommentare. Die Zusammenarbeit der Teilnehmer beschränkte sich auf die jeweilige Gruppe (*intra-group*). Die von den Gruppenmitgliedern annotierten Überprüfungsfragen wurden kaum im Sinne eines Selbsttests in Anspruch genommen, da dies eine erneute Auseinandersetzung mit dem selben Video bedeutet hätte. Eine Kollaboration innerhalb der Gruppen ist deshalb für Peer Annotation und Peer Assessment im Vergleich zur gruppenübergreifenden Zusammenarbeit [Seidel 2013] weniger geeignet.

Quellcode: <https://github.com/nise/vi-lab> (abgerufen am 22.01.2016).

Weiterführende Informationen: Seidel [2014c,d] (Ein Teil der Studien ist noch nicht abschließend ausgewertet und publiziert)

### 6.3.5 Theresienstadt explained

Gegenstand dieser Videolernumgebung ist der NS-Propagandafilm „Theresienstadt / Ein Dokumentarfilm aus dem Jüdischen Siedlungsgebiet“ aus dem Jahre 1944/45. Der von KZ-Häftlingen produzierte Film enthält als eines der wenigen historischen Zeitdokumente authentische Darstellungen der Schauplätze und Personen im Ghetto Theresienstadt. Aufgrund dieser Einzigartigkeit stellt der Film einerseits ein zentrales Lernmedium für die Schul- und Gedenkstättenpädagogik dar. Andererseits sind Aufführungen vor Gruppen mit der Herausforderung verbunden, die flüchtigen Informationen zu den Orten, Darstellern und den Stilmitteln der Propaganda mit notwendigen Kontextinformationen zu verknüpfen und somit die Propaganda aufzudecken. Die Anwendung *Theresienstadt explained* bietet unterstützende Funktionen für Filmvorführungen in Co-Präsenz. Darüber hinaus erhalten Kleingruppen die Möglichkeit, Szenen hinsichtlich auffälliger Täuschungen, Übertreibungen und Propaganda unter Verwendung eines CSCL-Scripts zu analysieren. Die jeweiligen Analyseergebnisse werden anschließend aggregiert und von einer anderen Gruppe kommentiert und bewertet.

Bisheriger Einsatz:

- Historische Einführung und kommentierte Aufführung anlässlich des *12. Neisse Filmfestivals* in der Hillerschen Villa Zittau.
- Online-Lernangebot (ohne CSCL-Script) unter <http://theresienstadt-film.net/> (abgerufen am 22.01.2016).

Quellcode: <https://github.com/nise/vi-theresienstadt> (abgerufen am 22.01.2016).

Zusammenfassend lässt sich konstatieren, dass das Hypervideo Framework *VI-TWO* sehr flexibel einsetzbar ist. Es ließ sich sowohl in bekannte Web 2.0 Werkzeuge wie dem *MediaWiki* und *WordPress*, als auch in komplexe Fallstudien wie *IWRM education*, *Theresienstadt explained* und nicht zuletzt in *VI-LAB* integrieren. Dabei wurde das Framework sukzessive gemäß der aus den *Interaction Design Patterns* resultierenden funktionalen Anforderungen erweitert und für jede Beispielanwendung getestet. Dieses iterative Vorgehen ging mit mehreren Überarbei-

tungen und einem intensiven *Refactoring* einher, um die Komplexität des Codes zu reduzieren und somit die Handhabung für Entwickler zu vereinfachen.

In Tab. 6.5 sind die in den zuvor skizzierten Anwendungen implementierten *Interaction Design Patterns* aufgeführt und als schwarze Punkte hervorgehoben. Aus dieser Aufstellung geht hervor, dass fast alle Patterns ein oder mehrmals umgesetzt wurden.

Die realisierten Patterns decken sich jedoch nicht vollständig mit dem auf S. 146 vorgestellten Entscheidungsmodell. In der Tabelle sind die nicht implementierten, jedoch notwendigen Patterns durch nicht ausgefüllte Kreise (○) gekennzeichnet. Die Gründe für diese Abweichung sind sehr unterschiedlich. In den ersten Anwendungen (*Vi-Wiki* und *IWRM education*) war das Framework noch nicht so weit entwickelt. Im Fall von *IWRM education* gab es zudem technische Anforderungen wie z.B. die serverunabhängige Nutzung auf portablen Speichermedien, die die Speicherung von Daten (z.B. ANNOTATIONS, USER NOTES) nicht gestattete. Hinzu kamen Vorgaben der Auftraggeber und zeitliche Einschränkung, die etwa mit dem Produktionsaufwand zur Erstellung zeitgenauer *tags* nicht vereinbar waren.

**Tabelle 6.5:** Die in den Anwendungen *Vi-Wiki*, *IWRM education* sowie den unterschiedlichen Entwicklungsstufen von *VI-LAB* implementierten (●) bzw. laut Entscheidungsmodell zur Implementierung empfohlenen (○) Interaction Design Patterns.

Interaction Design Pattern	Vi-Wiki	IWRM education	VI-LAB (v1)	VI-LAB (v2)	Theresienstadt
ADD VIDEO	●			●	
ANNOTATED TIMELINE	●	●	●	●	●
ANNOTATIONS	●	●	●	●	●
APPROPRIATE DELIVERY	○	○		○	●
ASSESSMENT			●	●	
BASIC CONTROLS	●	●	●	●	●
BRANCHING VIDEOS		●			
CLASSIFIED MARKS		●	●	●	●
CLOSED CAPTIONS		○			●
COMMENTS			●	●	
DETAIL ON DEMAND		●			
DIRECT AUTHORIZING	●		●	●	●
FOLLOW REVISIONS	●			○	
FULL SCREEN				○	●
HYPERLINKS	●	●		●	
JOURNALED NAVIGATION		●		○	●
LOADING INDICATOR	●	●	●	●	●
LOOP					●
MEDIA FRAGMENTS	○				●
MULTI-TIMELINE EDITING			●	●	●
OBJECT TRACKING					○
OVERLAYS	●		●	●	●
PLAYBACK SPEED		○		●	●
PLAYLIST		●			
RELATED VIDEOS		●	●		
REMIX	●				
SEARCH		●			○
SEQUENTIAL MEDIA	●	○			
SIMULTANEOUS MEDIA		●	●	●	●
SHORTCUT COMMANDS	●	●		●	●
SKIP BACK		●	○	●	●
SYNCHRONIZED MAP					●
TABLE OF CONTENT		●	●	●	●
TEMPORAL BOOKMARKS		●		●	●
TEMPORAL TAGS			●	●	●
TRANSCRIPT					●
USER NOTES		○			○
USER RATINGS		○			
USER TRACES		○			●
VIDEO MANAGER	●	●	●	●	●
VIDEO MANIPULATION					
VIEWING HISTORY		○		●	
VISUAL HIGHLIGHTING					●
VISUAL SUMMARY		○			●
ZOOM					●

## Zusammenfassung

In diesem Kapitel wurden mit *VI-TWO* und *VI-LAB* zwei generische Werkzeuge vorgestellt, die Lehrenden, Interaction Designern und Entwicklern die Konzeption und Entwicklung von Videolernumgebungen erleichtern.

*VI-TWO* ist ein modulares JavaScript-Framework zur Wiedergabe und Nutzung sowie Produktion von interaktiven Videos. Das Framework bietet erweiterte Steuerungsmöglichkeiten für den Videoplayer sowie Unterstützung für verschiedene Formen von Videoannotationen. In Bezug auf die Makrointeraktivität in einer Sammlungen aus mehreren Videos erhalten Nutzer die Möglichkeit in der Menge der Inhalte frei zu navigieren, zu suchen und sich die Lerninhalte individuell zu organisieren. Für die Produktion interaktiver Videos wurden typische Autorenwerkzeuge integriert. Der Mehrwert gegenüber ähnlichen Frameworks besteht in der Systemarchitektur, die aufgrund ihres modularen und erweiterbaren Aufbaus eine vielseitige Verwendung in verschiedenen Web 2.0 und E-Learning Anwendungen ermöglicht. Das Framework wurde bislang in fünf verschiedenartige Videolernumgebungen integriert und wurde dabei kontinuierlich weiterentwickelt. Einige Systeme befanden sich in mehrfachen und teilweise auch mehrjährigen Einsatz. 41 der 45 Patterns lassen sich mit dem Framework als Software Komponenten realisieren. Die Realisierung von Widgets gemäß der Patterns VIDEO MANIPULATION, OBJECT TRACKING, USER RATINGS und USER NOTES steht noch aus. Ausbaufähig sind zudem Anpassung für Smartphone- und Tablet-Apps wie *Firefox OS*, *Yun OS*, *Android* oder *iOS*.

*VI-LAB* als CSCL-System für videobasierte Scripts stand im Mittelpunkt des zweiten Teils des Kapitels. In der Lernumgebung wurden die in Abschnitt 2.2 (S. 17) spezifizierten Komponenten und Mechanismen videobasierter CSCL-Script größtenteils realisiert.

Das CSCL-System *VI-LAB* bietet Lehrenden die Möglichkeit videobasierte Scripts zu modellieren und sie in der virtuellen Lehre anzuwenden. Möglich wird dies vor allem durch die Integration von *VI-TWO*, dessen Bestandteile (Widgets) je nach Bedarf für Gruppen von Lernenden aktiviert und konfiguriert werden können. Zudem wird eine asynchrone und synchrone Kollaboration anhand von Videoannotationen und *thinkLets* ermöglicht.

Das System ist in der Lage, die in Teilabschnitt 5.1.4 (S. 178) beschriebenen Typen videobasierter CSCL-Scripts zu realisieren. Während die Durchführung verschiedener Varianten des PAPA-Scripts sowie jene zur Videoanalyse und -diskussion bereits Gegenstand von Studien im Rahmen von Lehrveranstaltungen waren, eröffnet das System noch sehr viel mehr Gestaltungsmöglichkeiten zur Strukturierung rezeptiver, selbstgesteuerter und kollaborativer Lernaktivitäten unterschiedlicher Komplexität. Insbesondere durch die verschiedenen Videoannotationen und die eingearbeiteten *thinkLets* lassen sich viele neuartige Szenarien für das Lernen mit Videos konzipieren. Mit *VI-LAB* sind darüber hinaus auch die technischen Voraussetzungen geschaffen, um weitere und vor allem gesicherte Erkenntnisse über videobasierte CSCL-Scripts gewinnen zu können. Die Durchführung und Auswertung von Feldstudien wird durch Monitoring-Funktionen, einfache Erhebungsinstrumente (Online-Fragebögen) und Datenschnittstellen zu den protokollierten Nutzeraktivitäten wesentlich erleichtert.

Auch wenn *VI-LAB* mit relativ vielen Features aufwartet und ein Lehrender zum „over-scripting“ [Dillenbourg 2002] geneigt sein könnte, lassen sich damit auch schlanke Scripts (Komplexitäts-

level 1) modellieren. Beipielsweise kann man regelmäßige Vorlesungsaufzeichnungen jeweils durch eine Script-Phase mit je einer Videoinstanz der Aufzeichnung definieren und der in einer Gruppe zusammengefassten Kohorte der Teilnehmer anbieten. Werden darüber hinaus Features wie die Suche oder zeitgenaue Kommentierung aktiviert, ist bereits ein Mehrwert gegenüber gewöhnlichen LMS geschaffen.

#### Kernpunkte

- Das Framework *VI-TWO* bildet 41 der 45 Interaction Design Patterns für Videolernumgebungen ab.
- *VI-TWO* wurde als Teil verschiedener Web-Anwendungen erprobt und hat sich aufgrund seines modularen Aufbaus als ein flexibles und erweiterbares Framework für Videolernumgebungen bewährt.
- Das CSCL-System *VI-LAB* ermöglicht die Modellierung und Anwendung videobasierter CSCL-Scripts, sowohl für komplexe kollaborative Lernszenarien als auch für das selbstgesteuerte und rezeptive Lernen mit Videos.
- *VI-LAB* unterstützt verschiedene Typen videobasierter CSCL-Scripts.
- In Bezug auf die Videoannotation erlaubt *VI-LAB* neben einer asynchronen, auch eine synchrone Kollaboration.





## Kapitel 7

# Schlussbetrachtung

In diesem letzten Kapitel der Arbeit wird ein kurzes Fazit gezogen, indem die Beiträge dieser Arbeit zur Forschung zusammengefasst und in Form einer kritischen Würdigung diskutiert werden. Am Ende des Kapitels wird ein Ausblick auf offengebliebene Forschungsfragen sowie weiterführende Forschungs- und Entwicklungsbedarfe gegeben.

### Beitrag der Arbeit zur Forschung

Diese Arbeit leistet einen Beitrag zur Forschung an Design Patterns sowie im Bereich Mensch-Computer-Interaktion und dem *Computer Supported Collaborative Learning*. Dazu wurden Methoden und Werkzeuge für die Gestaltung, den Einsatz und die Untersuchung videobasierter Lehr-Lern-Szenarien entwickelt. Folgende Beiträge sind im Wesentlichen zu nennen:

- Es wurde ein empirischer Ansatz für die Identifikation und Elaboration von *Interaction Design Patterns* vorgestellt und am Beispiel von Videolernumgebungen angewendet,
- Durch die Analyse von 121 Videolernumgebungen konnten Erkenntnisse über die Verbreitung und die Gestaltung einzelner Funktionalitäten und Gestaltungselemente gewonnen werden.
- Die 45 *Interaction Design Patterns* beschreiben häufige Lösungen für wiederkehrende Probleme bei der Gestaltung von Videolernumgebungen und stellen ein Hilfsmittel bei der Gestaltung, Entwicklung und Bewertung von Videolernumgebungen dar.
- Videobasierte CSCL-Scripts wurden als gesonderte Form von CSCL-Scripts eingeführt, anhand von Beispielen beschrieben und systematisch hinsichtlich ihrer Bestandteile und Mechanismen spezifiziert.
- Mit dem vorgestellten Software Framework *VI-TWO* lassen sich die in den *Interaction Design Patterns* beschriebenen Lösungen als interaktive Videos in verschiedenen CMS und LMS realisieren.
- Das implementierte CSCL-System *VI-LAB* gestattet es, videobasierte CSCL-Scripts zu modellieren und in Online und Blended Learning Szenarien einzusetzen.

## Kritische Würdigung

**Interaction Design Patterns für Videolernumgebungen** Mit dem vorgestellten Ansatz zur Identifikation und Elaboration von Interaction Design Patterns lässt sich nicht ausschliessen, dass gewisse Gestaltungslösungen nicht als Pattern erfasst wurden, da sie entweder bei der Inhaltsanalyse übersehen wurden oder sich in Videolernumgebungen wiederfinden, die nicht betrachtet wurden. Für die Inhaltsanalyse wurde deshalb ein zweiter Bewerter hinzugezogen (vg. S. 123). Die Wahrscheinlichkeit bestimmte Lösungen zu übersehen, wurde versucht durch die Einbeziehung einer großen Anzahl an Lernumgebungen zu reduzieren. Nichtsdestotrotz bedingt allein die technische und mediendidaktische Entwicklung in diesem Bereich, dass im Laufe der Zeit neue GUI-Lösungen emporkommen werden, die als potentielle Patterns zu prüfen sind. Aus diesem Grund stellt eine Pattern Language zunächst nur eine Momentaufnahme dar, mit welcher die Aufforderung zur kritischen Prüfung und Weiterentwicklung verbunden ist. Die veröffentlichte Wissensbasis ist deshalb als ein Werkzeug anzusehen, mit dem sich die Patterns einfach und schnell überarbeiten lassen und neue Lösungen als Beispiele integriert werden können.

Eine Einschränkung erfahren die Patterns hinsichtlich ihrer Anwendbarkeit. Sie stellen kontextspezifische Lösungen dar, die im Wesentlichen nur auf Web-Anwendungen und Apps für Mobilgeräte übertragbar sind. Anwendungen dieser Art zeichnen sich durch die Eingabe per Maus, Touchscreen oder Tastatur aus, während die Ausgabe auf verschiedenen großen Displays bzw. Projektionsflächen erfolgen kann. Die Patterns gelten somit nur bedingt für TV-Geräte, Spielkonsolen oder andere Geräte mit abweichenden Ein- und Ausgabemöglichkeiten. Beispielsweise würde eine videobasierte *Virtual Reality* Anwendung, die per Handgesten gesteuert und mit einem *head-mounted display* erfahrbar ist, ganz andere Anforderungen an das *User Interface* stellen und einen weitergefassten Gestaltungsraum beinhalten.

**CSCL-Scripts für Videolernumgebungen** Die vorgelegte Spezifikation für videobasierte CSCL-Scripts weist eine Reihe von Limitationen auf. Beispielsweise wurden obligatorische und fakultative Aufgaben von Tutoren und Dozenten während der Durchführung eines Scripts nicht berücksichtigt. Auch hinsichtlich des Detailgrads sind Weiterentwicklungen denkbar, um etwa Nutzerrollen innerhalb von Gruppen festzulegen oder Aufgaben nicht je Phase, sondern je Gruppe oder Teilnehmer zu vergeben. Ebenso könnte die Sichtbarkeit und der Aufgabenbezug einzelner Segmente, anstatt der gesamten Lernressource im Script spezifiziert werden. Gleiches gilt für Mechanismen wie das automatische Traversieren, Rotieren und Fading, die sich anhand von Templates noch genauer beschreiben lassen. Um letztendlich jedoch ein für den Anwender handhabbares Modell eines Scripts entwickeln zu können, sollte sich eine Spezifikation nicht in zu vielen Details erschöpfen und die Bedienung der Autorenumgebung verkomplizieren. Da die Spezifikation bislang nur aus einer relativ kleinen Anzahl bekannter videobasierter CSCL-Scripts abgeleitet werden konnte, bleibt abzuwarten, welche Erweiterungen und Reduktionen sich aus der Untersuchung neuer Scripts und ihrer praktischen Anwendung im Feld ergeben.

**Videolernumgebungen** Videolernumgebungen sind das zentrale Thema dieser Arbeit. Sie wurde als ein gesonderter Typus von Lernumgebungen postuliert. Die Untersuchung der 121 Videolernumgebungen unterstreicht diese Einschätzung im Hinblick auf das *User Interface De-*

*sign.* Die identifizierten Patterns stellen videospezifische Gestaltungslösungen dar, die in anderen Lernumgebungen in der Form nicht auftreten. Die Spezifikation der Script-Komponenten und Mechanismen hat jedoch gezeigt, dass sich videobasierte Lehr-Lern-Szenarien nicht zwingend von allgemeinen (script-gesteuerten) Szenarien im Blended und E-Learning unterscheiden müssen. Lediglich die mit dem Medium verknüpften und darin verankerten Aufgaben und Aktivitäten stellen ein Alleinstellungsmerkmal dar. In Abhängigkeit der didaktischen Intention des Videoeinsatzes (Beobachten, Erklären, Instruieren) und der technischen Repräsentation des Videos (z.B. Aufnahme realer Situationen vs. *Slidecasts*) sind Typen von Aufgaben umsetzbar, die sich auf das Prüfen von Faktenwissen, das Erkennen von Elementen oder Konzepten sowie das Treffen von Vorhersagen und die Anreicherung von Wissen beziehen können. Abgesehen von den Wissenstests sind diese Aufgabentypen im Zusammenhang mit Videos bislang kaum verbreitet und nur ansatzweise erforscht [Seidel 2016].

Die im Rahmen dieser Arbeit erfolgten softwaretechnischen Neuentwicklungen von *VI-TWO* und *VI-LAB* haben sich im praktischen Einsatz mehrfach bewährt und sind über mehrere Jahre gereift. Nichtsdestotrotz muss betont werden, dass es sich bei *VI-LAB* vorerst nur um einen Prototypen handelt, der für den Einsatz als Produktivsystem an verschiedenen Stellen ausgebaut werden müsste. Beispielsweise können derzeit noch nicht mehrere Scripts gleichzeitig ausgeführt werden. Die Performanz des Systems wurde bislang nur für den Betrieb mit weniger als 100 Anwendern getestet.

## Ausblick

**Design Patterns** Die vorgestellten Patterns fokussieren das Interaction Design. Während Software Entwickler zumindest durch das vorgestellte Framework *VI-TWO* noch eine gewisse Unterstützung bei der Umsetzung erfahren, genügen die in den Patterns teils enthaltenden mediendidaktischen und kognitionspsychologischen Anknüpfungspunkte nicht aus, um daraus schlüssige Lernszenarien zu entwickeln. Der Frage wie didaktische Handlungsempfehlungen oder Patterns für das videobasierten Lehren und Lernen aussehen könnten, wurde bislang nur ansatzweise nachgegangen (z.B. Mor und Warburton [2016]; Wannemacher [2016]).

**Videobasierte CSCL-Scripts** Der Kenntnisstand zu videobasierten CSCL-Scripts ist gegenwärtig noch sehr gering. Zwar wurde in dieser Arbeit versucht die Struktur der Scripts zu beschreiben, jedoch ist die Wirksamkeit und Akzeptanz bislang kaum belegt. Die vom Autor durchgeführten Feldstudien [Seidel 2013, 2014d] sind aufgrund zu geringer Fallzahlen (< 100 Personen) nicht repräsentativ. Es besteht deshalb ein hoher Forschungsbedarf, einzelne Scripts und Script-Varianten in Bezug auf unterschiedliche Zielgruppen, Fachbereiche und Videolernressourcen empirisch zu untersuchen. Da komplexe Scripts Lernende und Lehrende überfordern können, stellt sich auch die Frage, welche Scripts sich auf welche Weise vereinfachen lassen.

**VI-TWO und VI-LAB** Das CSCL-System *VI-LAB* ist grundsätzlich flexibel und mächtig genug, um als Untersuchungsumgebung für eine Vielzahl von Feldstudien zu dienen. Auch für den regelmäßigen Einsatz in der Lehre ist es, wie die angeführten Beispiel auf S. 248 zeigen, geeignet. Dennoch handelt es sich bei der Anwendung noch um einen Prototypen, der nicht vollständig für die Zielgruppe der Lehrenden optimiert wurde. Eine Untersuchung der Akzep-

tanz (z.B. [Venkatesh et al. 2003]) und Usability unter Berücksichtigung der *User Experience* (z.B. [Rauschenberger et al. 2013]) steht noch aus.

Forschungs- und Entwicklungsbedarf besteht auch hinsichtlich der Interoperabilität. Eine Entwicklung von einheitlichen Standards für Hypervideos und CSCL-Scripts, etwa durch Arbeitsgruppen beim *World Wide Web Consortium* bzw. beim *IMS Global Learning Consortium* erscheint deshalb sehr erstrebenswert. Die bisherigen Bemühungen einen *Open Video Capture Standard*<sup>20</sup> (abgerufen am 8.11.2016) unter dem Dach von *IMS Global Learning Consortium* zu definieren, berücksichtigen zumindest Videoannotationen in Form von synchronisierten Medien und Inhaltsverzeichnissen. Mit Scripts vergleichbare Settings lassen sich damit jedoch nicht beschreiben.

Auch hinsichtlich *Multi-angle Videos*, sphärischen Videos (z.B. 360°Videos) sowie *Virtual Reality* Anwendungen für das Lehren und Lernen mit Videos besteht ein Forschungs- und Entwicklungsbedarf, da die dazu erforderliche Hard- und Software auch für Hochschulen erschwinglich geworden ist und entsprechende Anwendungsszenarien seit längerem bekannt sind (z.B. Wallbaum [2010]; Zahn et al. [2012]).

**Beziehung von Patterns und Scripts** Die Hauptwerke zur Scripttheorie [Schank und Abelson 1977] und Mustertheorie [Alexander et al. 1977] gehen auf das gleiche Jahr zurück. Obwohl beide Ansätze Schwerpunkt dieser Arbeit sind, wurden bislang nur wenige Querbeziehungen hergestellt. Scripts und Design Patterns sind Wissensrepräsentationen, welche man aus realen, häufig wiederkehrenden Situationen abstrahiert, um sie mittels ihrer Beschreibung zur Gestaltung ebensolcher Situationen zu nutzen. Der Unterschied besteht vor allem in Form und Inhalt der Beschreibung. Scripts enthalten beispielsweise keine Problembeschreibung, Forces oder Beziehungen zu anderen Patterns bzw. Scripts. Im Unterschied zu Patterns sind CSCL-Scripts jedoch konkrete, ausführbare Festlegungen von Interaktionen, die Individuen und Gruppen mit dem System und den Lernressourcen zur Erreichung eines Lernziels vollführen.

Beide Ansätze sind jedoch miteinander kombinierbar. Wie bei der Spezifikation und Modellierung videobasierter CSCL-Scripts gezeigt wurde, lassen sich Teile von CSCL-Scripts als eine Implementierung von Interaction Design Patterns umsetzen. Umgekehrt lässt sich der Dreiklang von Problem, Lösung und Kontext jedoch auch für die Modellierung von Scripts einsetzen [Hernandez-Leo et al. 2006]. Die Frage lautet daher: Welche erprobten Konfigurationen von Script Komponenten und Mechanismen haben sich für wiederkehrende Probleme bei der Gestaltung von kollaborativen Lehr-Lern-Szenarien mit Videos bewährt? *Jigsaw*-Scripts könnten unter Umständen einen Pattern bilden, in dem die Auswahl eines bestimmten Verfahrens zur Gruppenbildung und Verteilung von Lernressourcen verbunden ist. Der Forschungsbedarf besteht nun darin, in der Praxis bewährte Konfigurationen von videobasierten CSCL-Scripts zu identifizieren und sie als Pattern-Kandidaten zu dokumentieren.

---

<sup>20</sup>Siehe <https://www.imsglobal.org/OpenVideoCallforParticipation.html>

**Anhang A**

**Anhang**

## Übersicht wissenschaftlicher Konferenzen und Sonderausgaben von Journalen zum Lernen mit Videos

Kategorie	Datum	Veranstaltung	Hochschule/Ort, Land	Link
Workshop	13.09.2005	DeLFI Workshop VAZ	Rostock, Deutschland	–
Konferenz	05./06.09.2012	Lernen mit (Online-)Videos	Universität Bremen, Deutschland	<a href="http://blogs.uni-bremen.de/lernenmitvideo/symposium/">http://blogs.uni-bremen.de/lernenmitvideo/symposium/</a>
Workshop	08.04.2013	Workshop on Analytics on Video-based Learning	Leuven, Belgien	<a href="https://sites.google.com/a/ionio.gr/wave/">https://sites.google.com/a/ionio.gr/wave/</a>
Konferenz	18.09.2013	12. E-Learning Tag 2013 - Lernen mit Videos und Spielen	FH Joanneum / Graz, Österreich	<a href="https://sites.google.com/site/elearningtag2013/home">https://sites.google.com/site/elearningtag2013/home</a>
Teilkonferenz	04.-06.12.2013	VideoEduca, OnlineEduca	Berlin, Deutschland	–
Sonderheft	01.04.2014	„Videos in der (Hochschul-)Lehre“, Zeitschrift für Hochschulentwicklung	Österreich	<a href="http://www.zfhe.at/index.php/zfhe/issue/view/41">http://www.zfhe.at/index.php/zfhe/issue/view/41</a>
Workshop	14.03.2014	AV @ GML <sup>2</sup>	FU Berlin, Deutschland	<a href="http://www.gml-2014.de/pre-conference/audio-video.html">http://www.gml-2014.de/pre-conference/audio-video.html</a>
Workshop	09-10.05.2014	10. Forschungswerkstatt „Social Video Learning“ - Eine didaktische Expedition	Donau-Universität Krems, Österreich	<a href="http://www.donau-university.ac.at/de/departments/imb/veranstaltungen/id/20685/">http://www.donau-university.ac.at/de/departments/imb/veranstaltungen/id/20685/</a>
Teilkonferenz	03.-05.12.2014	VideoEduca, OnlineEduca	Berlin, Deutschland	–
Teilkonferenz	02.-04.12.2015	VideoEduca, OnlineEduca	Berlin, Deutschland	–

## Erweiterte Usability Guidelines für interaktive Videos

**Vorbemerkung** Die in den Tabellen A.1, A.2, A.3, A.4 und A.5 dargestellten Guidelines basieren auf den Vorarbeiten von Meixner et al. [2011]. Spezielle Aspekte hinsichtlich zeitabhängiger Objekte und Bedienelemente (Tab. A.6) und der barrierefreien Gestaltungen (Tab. A.7) wurden ergänzt.

**Anwendung der Guidelines** Jede Aussage in den sieben Tabellen, die für die betrachtete Anwendung relevant ist, wird mit einer Wertung von 1 bis 5 Punkten versehen. Ist eine Aussage gänzlich erfüllt, werden 5 Punkte vergeben. Wird sie hingegen nicht erfüllt, werden 0 Punkte erteilt. Falls keine Aussage möglich ist, fällt die Frage aus der Bewertung raus.

**Quantifizierung der Usability** Sei  $A = \{a_1, a_2, \dots, a_{77}\}$  die Menge aller in den Guideline enthaltenen Aussagen, wobei  $|A| = 77$ . Sei ferner  $R = \{r_1, r_2, \dots, r_n\} \mid 0 \leq r \leq |A|$  eine Teilmenge  $R \subseteq A$  derjenigen Aussagen, die in der betrachteten Anwendung auch umgesetzt wurden. Jede bewertete Aussage  $r_i \in R$  ist eine Wertung  $w_i \in W \mid W = \{0, 1, 2, 3, 4, 5\}$  mit  $W \in \mathbb{N}$  zugeordnet, wobei  $f : R \rightarrow W$ .

Der Grad der Bedienbarkeit  $u \in \mathbb{R}$ , ( $0 \leq u \leq 1$ ) einer betrachteten Anwendung  $P$  berechnet sich aus dem Verhältnis der Summe aller abgegebenen Wertungen zur maximal erreichbaren Wertung bezüglich der ausgewählten Aussagen:

$$u_P = \frac{\sum_{r_i \in R} w_i}{5 * |R|} \quad (\text{A.1})$$

### Quellen

- Travis, D. (2014). 247 web usability guidelines. <http://www.userfocus.co.uk/resources/guidelines.html> (abgerufen am 22.11.2015).
- U.S. Department of Health and Human Services & U.S. General Services Administration (2015). Research-Based Web Design & Usability Guidelines. <http://guidelines.usability.gov/> (abgerufen am 22.11.2015).
- Bevan, N. & Kincla S. (2004). HCI Design Foundation Study. Final Report. 1.0, Serco Usability Services, Serco Limited for JISC, London, UK.
- Caldwell, B., Cooper, M., Reid, L. G., Vanderheiden, G., Chisholm, W., Slatin, J., & White, J. (2008). Web Content Accessibility Guidelines (WCAG) 2.0. <http://www.w3.org/TR/WCAG20/> (abgerufen am 22.11.2015).



**Tabelle A.1: Design:** Durch die Farbgestaltung und das Layout der einzelnen Bereiche und des gesamten Players wird beschrieben, wie Elemente angeordnet und beschriftet, beziehungsweise mit Icons versehen werden sollen. Ein weiterer Fokus liegt auf dem Erkennen klickbarer Elemente. Es wird beschrieben, wie sich Elemente verhalten sollen. Der gesamte Aufbau und die Bereiche sollen zudem selbsterklärend sein.

Nr.	Beschreibung	Quelle	Wert
1.1	Positiver erster Eindruck durch ansprechendes Design.	[HHS 2015] ,[Travis 2014](2x)	
1.2	Der Player ist selbsterklärend.	[Travis 2014] (7x)	
1.3	Der Aufbau/die Struktur des Players ist einfach und besitzt ein klares Konzept.	[Travis 2014]	
1.4	Verwendung geeigneter Farbschemata und Hintergründe.	[HHS 2015] ,[Travis 2014]	
1.5	Kontrastreiche Darstellung von Text zu Hintergrund.	[HHS 2015] , HCI	
1.6	Aufteilung des Players in klar definierte Bereiche.	HCI	
1.7	Wichtige (Steuer-)Elemente werden an gut erreichbarer und geeigneter Stelle platziert.	[HHS 2015] (2x), [Travis 2014](6x), HCI	
1.8	Zusammengehörende Elemente und Buttons wurden in räumlicher Nähe angeordnet.	UF	
1.9	Konsistente Gestaltung zusammengehörender Elemente.	[HHS 2015] (2x),[Travis 2014](2x)	
1.10	Es besteht ein klarer Zusammenhang zwischen Labels/Icons eines Buttons und dessen Funktion.	[HHS 2015] , [Travis 2014](4x)	
1.11	Wichtige Funktionen werden über Buttons gesteuert.	[Travis 2014] (2x)	
1.12	Verwendung von Icons die Objekten der echten Welt entlehnt sind und einen hohen Wiedererkennungswert haben.	[HHS 2015] ,[Travis 2014]	
1.13	Terminologie und Konventionen von Links sind mit Weblinks konsistent.	[Travis 2014] (3x), HCI, [HHS 2015]	
1.14	Nicht klickbare Elemente unterscheiden sich von klickbaren (v.A. bei Bildern).	[Travis 2014] (2x), HCI, [HHS 2015] (2x)	
1.15	Bereits geklickte Elemente sind von den noch nicht geklickten unterscheidbar.	[Travis 2014]	
1.16	Vermeidung von <i>Rollover</i> - und <i>Pulldown-Menüs</i> , stattdessen Verwendung von <i>Point and Click</i> .	HCI, [HHS 2015]	
1.17	Der Mauszeiger verändert sich wenn er sich über klickbaren Elementen befindet.	[Travis 2014]	
1.18	Verwendung von gut lesbaren Schriftarten in angemessener Größe.	[HHS 2015] (2x),[Travis 2014]	
1.19	Bei Texten ist die Zeilenlänge angemessen.	[Travis 2014]	
1.20	Vermeidung von <i>Scrollbars</i> .	HCI,[Travis 2014](2x), [HHS 2015]	
1.21	Vermeidung von Pop-Ups.	[HHS 2015]	
1.22	Bereitstellen von Vorschau-Elementen oder Texten bei Mediendateien und Bildern.	[HHS 2015] (2x), HCI	

**Tabelle A.2: Navigation:** Der Aufbau eines nicht-linearen Videos kann bis zu einem gewissen Grad mit dem Aufbau einer Webseite verglichen werden. Deshalb sind Elemente für die Navigation im Video, wie die Anzeige der aktuellen Position (soweit dies in einem nicht-linearen Video ohne feste Gesamtdauer möglich ist) nötig.

Nr.	Beschreibung	Quelle	Wert
2.1	Wichtige Funktionen und Bereiche des Players sind immer erreichbar.	[Travis 2014](3x)	
2.2	Anzeige der aktuellen Position im Video.	[HHS 2015] (2x), [Travis 2014]	
2.3	Verwendung von <i>Breadcrumbs</i> .	[HHS 2015] , HCI	
2.4	Verwendung einer geeigneten und immer erreichbaren Video-Map (Szenengraph).	[HHS 2015] ,[Travis 2014](3x)	
2.5	Vor- und Zurück-Button zum Sprung zwischen Szenen.	HCI	

**Tabelle A.3: Suche:** Unterstützt der Videoplayer eine Suchfunktion, sind bezüglich der Gestaltung dieser grundlegende Richtlinien zu beachten. Die Suchfunktion soll zum einen den Benutzer bei der Eingabe der Suchbegriffe unterstützen, zum anderen müssen die Suchergebnisse in geeigneter Weise aufbereitet werden. Eine funktionale Aufbereitung der Suchergebnisse erhöht die Bedienbarkeit.

Nr.	Beschreibung	Quelle	Wert
3.1	Das Suchfeld wird immer an gleicher Stelle platziert (wo der Betrachter es vermutet).	HCI,[Travis 2014](2x)	
3.2	Die Suchfunktion unterstützt den Benutzer bei der Eingabe der Suchbegriffe.	[Travis 2014] (2x)	
3.3	Das Eingabefeld für Suchwörter ist ausreichend groß für übliche Eingaben.	HCI,[Travis 2014]	
3.4	Die Suchfunktion erweitert die Suchbegriffe intelligent, um effektivere Ergebnisse zu erhalten.	[Travis 2014], [HHS 2015]	
3.5	Ausgabe einer sinnvollen Seite, falls die Suche keine Ergebnisse liefert.	[Travis 2014] (2x)	
3.6	Es wird der gesamte Index des interaktiven Videos durchsucht.	[Travis 2014], [HHS 2015]	
3.7	Geeignete Aufbereitung der Suchergebnisse durch Anzeige von Meta-informationen, Gruppierung von Listen, Sortierung nach Relevanz, etc.	HCI (2x),[Travis 2014](5x), [HHS 2015] (3x)	
3.8	Oft gesuchte Begriffe bringen nützliche Suchergebnisse.	[Travis 2014]	
3.9	Optionale Kontrollelemente ermöglichen eine Verfeinerung der Suchergebnisse.	HCI,[Travis 2014](3x)	
3.10	Es wird eine Ähnlichkeitssuche angeboten.	[Travis 2014]	
3.11	Der Player unterstützt sowohl die Suche als auch ein Inhaltsverzeichnis.	[Travis 2014]	

**Tabelle A.4: Technologie:** Implementierungskonzepte für Teilbereiche des Players. Eine automatische Anpassung an verschiedene Endgerätespezifikationen ist besonders bei Videos wichtig, um lange Wartezeiten oder schlechte Abspielqualität zu vermeiden. Ebenso sind Vorgehensweisen und Prinzipien der Software Entwicklung einzuhalten.

Nr.	Beschreibung	Quelle	Wert
4.1	Automatische Anpassung an allgemein verwendete Endgerätespezifikationen (Bildschirmauflösung, Browser, Dateiformate, Betriebssysteme, Datenrate).	[HHS 2015] (7x),[Travis 2014]	
4.2	Synchronisation von Multimedia-Elementen.	[HHS 2015]	
4.3	Der Player ist robust und alle Schlüsselfunktionen funktionieren.	[Travis 2014]	
4.4	Funktionen entsprechen den Erwartungen der Benutzer und richten sich an dessen Aufgaben/Möglichkeiten aus.	[HHS 2015] (2x),[Travis 2014]	
4.5	Es wird sichergestellt, dass keine Daten verloren gehen (z.B. Kommentare, Bookmarks, kollaborative Daten).	[Travis 2014]	
4.6	Interne Abläufe des Players bleiben dem Benutzer verborgen.	[Travis 2014]	
4.7	Die Wiedergabe ist ohne eine Installation von Zusatzprogrammen möglich.		

**Tabelle A.5: Benutzerfeedback und Fehlermeldungen:** Tritt ein Fehler auf, muss der Benutzer darüber in angemessener Art und Weise informiert werden. Sollte es zum Beispiel durch lange Downloadzeiten zu Problemen oder Verzögerungen im normalen Ablauf des Videos kommen, muss der Benutzer darüber informiert werden. Hier sind Guidelines angeführt, die bezüglich des Benutzerfeedback eingehalten werden müssen.

Nr.	Beschreibung	Quelle	Wert
5.1	Konstruktive und höfliche Formulierung von Fehlermeldungen.	[Travis 2014]	
5.2	Verwendung von Fehlermeldungen mit Bezug zum Problem.	[Travis 2014], HCI	
5.3	Eine Fehlermeldung wird angezeigt, sobald eine Annotation oder eine Szene nicht geladen wurde.	[Travis 2014], HCI	
5.4	Bei längeren Lade- und Wartezeiten gibt der Player Rückmeldung über den Ladefortschritt.	[Travis 2014] (2x)	
5.5	Tooltips geben nützliche Hinweise über die Beschriftung oder Symbolik der Buttons hinaus.	[Travis 2014]	
5.6	Angebot von Hilfestellung für den Benutzer, sofern notwendig.	[HHS 2015]	

**Tabelle A.6: Zeitabhängige Objekte und Bedienelement:** Zeitabhängige Navigations- und Bildelemente innerhalb des Videobilds stellen eine besondere Herausforderung dar, da sie das Bewegtbild überlagern und aufgrund ihrer zeitlich begrenzten Sichtbarkeit bedienbar sein müssen. Die folgenden Guidelines adressieren Aspekte der Bedienung von zeitabhängigen Elementen sowie anderen Darstellungen im Videobild.

Nr.	Beschreibung	Quelle	Wert
6.1	Zeitabhängige Bedienelemente sind als solche erkennbar und deutlich vom Hintergrund des Bewegtbildes unterscheidbar.	–	
6.2	Zeitabhängige Objekte (Overlays) verdecken keine relevanten Informationen im Videobild.	–	
6.3	Zeitabhängige Objekte (Overlays) überlappen sich nicht gegenseitig im Videobild.	–	
6.4	Die Anzahl der gleichzeitig dargestellten zeitabhängigen Objekte und Bedienelemente im Videobild ist angemessen.	–	
6.5	Es wird ersichtlich, wann eine Annotation (z.B. Hyperlink) im Video erscheinen wird.	–	
6.6	Die zeitliche Verteilung von Ereignissen im Verlauf der Wiedergabezeit (insb. Annotationen) ist nachvollziehbar.	–	
6.7	Zeitliche Ergebnisse sind unabhängig von der aktuellen Abspielposition direkt zugänglich.	–	
6.8	Die Anzeigedauer zeitabhängiger Objekte und Bedienelemente ist ausreichend, um sie zu rezipieren oder sie ggf. zu aktivieren.	–	
6.9	Es besteht die Möglichkeit die Anzeigedauer zeitabhängiger Objekte und Bedienelemente zu verlängern.	–	
6.10	Die Darstellung von zeitabhängigen Bedienelementen suggeriert die Folgen ihrer Aktivierung.	–	
6.11	Nach Aktivierung eines zeitabhängigen Bedienelements besteht die Möglichkeit zum Video zurückzukehren und die Wiedergabe fortzusetzen.	–	
6.12	Es besteht die Möglichkeit, entlang des Verlaufspfades der zuletzt betrachteten Videos zu navigieren.	–	

**Tabelle A.7: Barrierefreiheit:** Die Herausforderung einer barrierefreien Gestaltung der Schnittstelle zwischen Anwender und Video resultiert maßgeblich aus der kritischen Zugänglichkeit des Medium Video selbst. Schwierigkeiten ergeben sich damit für Menschen, die physischen oder psychischen Einschränkungen unterliegen. Die Art und Weise, in der Menschen bei der Benutzung technischer Systeme eine Benachteiligung erfahren können, ist dabei sehr unterschiedlich. Betroffen davon sind diverse Personengruppen in unterschiedlichen Lebenssituationen. Im Kontext des E-Learnings betrifft dies nicht nur Lernende mit verminderter Seh- oder Hörleistung, sondern auch Menschen mit einer langsamen Reaktionsgeschwindigkeit oder etwa Personen, die fremdsprachlichen Lerninhalten rezipieren.

Nr.	Beschreibung	Quelle	Wert
7.1	Die Steuerung des Videoplayers ist auch per Tastatur möglich.	Caldwell et al. [2008, Guideline 2.1.1]	
7.2	In allen Fallback-Varianten des Players sind die Bedienelemente in gleicher Weise zugänglich.	Caldwell et al. [2008, Guideline 3.2.3]	
7.3	Videos können in verschiedenen Auflösungen und Qualitätsstufen betrachtet werden.	Caldwell et al. [2008, Guideline 1.1.1]	
7.4	Die Wiedergabegeschwindigkeit ist anpassbar.	–	
7.4	Der Player ist mit einem Screenreader (z.B. JAWS) oder Braille-Ausgabegerät nutzbar.	Caldwell et al. [2008, Guideline 1.3.1]	
7.5	Die Lautstärke des Videos ist regelbar.	Caldwell et al. [2008, Guideline 1.4.2]	
7.6	Für das Video ist eine Zusammenfassung der Audioinformationen verfügbar.	Caldwell et al. [2008, Guideline 1.2.1/3]	
7.7	Für das Video ist eine ausführliche Beschreibung der Audioinformationen verfügbar (Extended Audio Description).	Caldwell et al. [2008, Guideline 1.2.1/3]	
7.8	Für das Video wird ein Transkript angeboten.	Caldwell et al. [2008, Guideline 1.2.1/3]	
7.9	Untertitel werden in einem gängigen Format angeboten (z.B. Timed Text, QTtext/QuickTime text, SubRip/.srt)	Caldwell et al. [2008, Guideline 1.2.1]	
7.10	Untertitel können wahlweise oder permanent eingeblendet werden. (closed captions/cc vs. open captions)	Caldwell et al. [2008, Guideline 1.2.2]	
7.11	Die Untertitel stehen in mehreren Sprachen zur Verfügung.	Caldwell et al. [2008, Guideline 1.2.1/2]	
7.12	Es steht eine zusätzliche Audiospur in einer anderen Sprache zur Verfügung.	Caldwell et al. [2008, Guideline 1.2.1]	
7.13	Ein Gebärdensprachvideo steht als alternative Aufbereitung zur Verfügung.	Caldwell et al. [2008, Guideline 1.2.1]	

## Unterstützung gängiger Video-Codecs durch verbreitete Web-Browser

**Tabelle A.8:** Unterstützung gängiger Video-Codecs durch verbreitete Web-Browser

Browser	Version	Engine	H.264 (MP4)	Matros- ka / VP8 / VP9 (WebM)	ogg theora (ogv)	wmv
Google Chrome/Chromium	28.x2	Blink, V8	●	●	●	-
Mozilla Firefox	22.x	Gecko, SpiderMonkey	●	●	●	-
Microsoft Internet Explorer	11.x	Trident, Chakra	●	-	-	●
Opera	12.x	Blink, V8	●	●	●	-
Apple Safari	8.x	WebKit, nitro	●	-	-	-

Die folgenden Browser unterstützen das HTML5 Video-Element:

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Blackberry Browser	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android
	32	37	5	24	4.1		3						
6	33	38	5.1	25	4.3		4						
7	34	39	6	26	5.1		4.1						
8	35	40	6.1	27	6.1		4.3						
9	36	41	7	28	7.1		4.4	12					
10	37	42	7.1	29			4.4.4	12.1	7			10	
11	38	43	8	30	Opera 30 Supported		40	24	10	42	38	11	9.9
Edge	39	44	9	31	Browser usage Global: 0.01%								
	40	45		32									
	41	46											

**Abbildung A.1:** Übersicht über die Browser-Versionen, die das HTML5 Video-Element unterstützen.  
Quelle: Can I Use, <http://caniuse.com/#feat=video> (abgerufen am 15.06.2015).

## Recherchedatenbanken und -verzeichnisse

**Tabelle A.9:** Spezifische Suchbegriffe je Rechercheplattform

Kategorie	Rechercheplattform	Suchberiffe
Fachdatenbank	<a href="http://sciencedirect.com/">http://sciencedirect.com/</a>	video instruction; video learning; video environment; video player; hypervideo; interactive video;
Fachdatenbank	<a href="http://dl.acm.com/">http://dl.acm.com/</a>	hypervideo; video learning;
Fachdatenbank	<a href="http://link.springer.com/">http://link.springer.com/</a>	video instruction; video learning; video environment; video player; hypervideo; interactive video;
Fachdatenbank	<a href="http://dl.gi.de/">http://dl.gi.de/</a>	hypervideo; interaktives video; video lernen;
Fachdatenbank	<a href="https://archive.org/">https://archive.org/</a>	hypervideo; interactive video;
Kurznachrichtendienst	<a href="http://twitter.com/">http://twitter.com/</a>	hypervideo; interactive video;
Research Network	<a href="http://mendeley.com/">http://mendeley.com/</a>	hypervideo; video learning; interactive video; audiovisual learning;
Social Bookmarking	<a href="http://delicio.us/">http://delicio.us/</a>	vidcast learning; hypervideo; video collaboration; video interactive; video learning 2.0;
Social Bookmarking	<a href="http://digg.com/">http://digg.com/</a>	video; video elearning; video interactive; hypervideo;
Suchmaschine	<a href="https://www.google.com/alerts/">https://www.google.com/alerts/</a>	interactive video -game; hypervideo; video learning environment;
Suchmaschine	<a href="http://scholar.google.com/">http://scholar.google.com/</a>	interactive video -game; hypervideo; video learning environment; video learning;
Suchmaschine	<a href="http://books.google.com/">http://books.google.com/</a> , <a href="https://books.google.com/ngrams/">https://books.google.com/ngrams/</a>	interactive video -game; hypervideo; video learning environment; video learning;
Firmendatenbank	<a href="https://info.crunchbase.com/">https://info.crunchbase.com/</a>	video;



## Verzeichnis der untersuchten Videosysteme

### – **56.com**

Provider: 56.com

URL: <http://www.56.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment

Patterns: Add Video, Loading Indicator, Playlist, Related Videos, Temporal Bookmarks, Viewing History, Media Fragments, Video Manipulation, Basic Controls

Funktionalitätsrate: 0.18

### – **ADIVI**

Provider: InnoTeamS GmbH

URL: <http://www.adivi.net/en/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment

Patterns: Add Video, Related Videos, Detail on Demand, Simultaneous Media, Hyperlinks, Overlays, Visual Highlighting, Branching Videos, Annotations, Basic Controls

Funktionalitätsrate: 0.2

### – **Adobe Captivate**

Provider: Adobe Inc.

URL: <http://www.adobe.com/de/products/captivate.html> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Authoring Environment, Desktop Environment

Patterns: Add Video, Appropriate Delivery, Assessment, Basic Controls

Funktionalitätsrate: 0.08

### – **Advene**

Provider: LIRIS, University of Lyon

URL: <http://liris.cnrs.fr/advene/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Desktop Environment, Annotation Tool, Open Source

Patterns: Add Video, Annotations, User Notes, Object Tracking, Annotated Timeline, Transcript, Closed Captions, Multi-Timeline Editing, Multi-Timeline Editing, Basic Controls

Funktionalitätsrate: 0.2

### – **Amara / Universal Subtitles**

Provider: Participatory Culture Foundation

URL: <http://universalsubtitles.org/en/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Annotation Tool

Patterns: Annotated Timeline, Annotations, Closed Captions, Shortcut Commands, Direct Authoring, Multi-Timeline Editing, Basic Controls

Funktionalitätsrate: 0.14

### – **Amazonie**

Provider: ARTE GEIE

URL: <http://amazonie.arte.tv/de/> (abgerufen am 15.01.2016)

Gruppierung: Interactive Film, Online Environment

Patterns: Related Videos, Annotated Timeline, Overlays, Hyperlinks, Sequential Media, Closed Captions, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.16

### – **Annotating Academic Video**

Provider: ETH Zürich, SWITCH, University of Bern, PHZH and others

URL: <https://github.com/entwinemedia/annotations> (abgerufen am 15.01.2016)

Gruppierung: Online Environment, Annotation Tool, Open Source

Patterns: Annotations, User Notes, Annotated Timeline, Classified Marks, Comments,

- Multi-Timeline Editing, Direct Authoring, Basic Controls  
Funktionalitätsrate: 0.16
- **Apreggio**  
Provider: Viddler, Inc  
URL: <http://info.viddler.com/arpeggio-features> (abgerufen am 15.01.2016)  
Gruppierung: Player, Online Environment  
Patterns: Comments, Annotated Timeline, Overlays, Playback Speed, Table of Content, Assessment, Closed Captions, Annotations, Basic Controls  
Funktionalitätsrate: 0.18
  - **Arte.tv**  
Provider: ARTE G.E.I.E.  
URL: <http://videos.arte.tv/de/videos/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment  
Patterns: Search, Video Manager, Temporal Bookmarks, Full Screen, Playlist, Related Videos, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.16
  - **Asterpix**  
Provider: Asterpix (has disappeared)  
URL: (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Add Video, Hyperlinks, Overlays, Object Tracking, Annotated Timeline, Detail on Demand, Annotations, Basic Controls  
Funktionalitätsrate: 0.16
  - **BMRC Lecture Browser**  
Provider: Berkeley Multimedia Research Center (BMRC)  
URL: <http://www.eecs.berkeley.edu/XRG/Summary/Old.summaries/01abstracts/chema.1.html> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Authoring Environment, Online Environment, Desktop Environment, Research Prototype  
Patterns: Search, Hyperlinks, Temporal Tags, Simultaneous Media, Table of Content, Annotated Timeline, Basic Controls  
Funktionalitätsrate: 0.14
  - **BlackGoldBoom**  
Provider: Todd Melby (lead producer)  
URL: <http://blackgoldboom.com/> (abgerufen am 15.01.2016)  
Gruppierung: Interactive Film, Online Environment  
Patterns: Sequential Media, Hyperlinks, Overlays, Branching Videos, Basic Controls  
Funktionalitätsrate: 0.1
  - **BubblePLY**  
Provider: PLYmedia Inc.  
URL: <https://www.youtube.com/watch?v=aXBYJ0ZdEqM> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Add Video, Overlays, Hyperlinks, Annotations, Direct Authoring, Closed Captions, Detail on Demand, Direct Authoring, Basic Controls  
Funktionalitätsrate: 0.18
  - **Clipfish**  
Provider: Clipfish GmbH & Co. KG  
URL: <http://www.clipfish.de/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment

- Patterns: Add Video, Playlist, User Ratings, Video Manager, Search, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.14
- **Clouds over Cuba**  
Provider: John F Kennedy Presidential Library and Museum  
URL: <http://cloudsovercuba.com/> (abgerufen am 15.01.2016)  
Gruppierung: Interactive Film, Online Environment  
Patterns: Classified Marks, Playback Speed, Overlays, Sequential Media, Branching Videos, Short-cut Commands, Annotated Timeline, Hyperlinks, Basic Controls  
Funktionalitätsrate: 0.18
  - **Control Video Speed**  
Provider:  
URL: <http://www.controlvideospeed.com/> (abgerufen am 15.01.2016)  
Gruppierung: Desktop Environment, Player  
Patterns: Playback Speed, Basic Controls  
Funktionalitätsrate: 0.04
  - **CyTube**  
Provider: Calvin Montgomery  
URL: <http://www.synchtube.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment, Open Source  
Patterns: Playlist, Basic Controls  
Funktionalitätsrate: 0.04
  - **DIVER/WebDIVER**  
Provider: Stanford Center for Innovations in Learning (SCIL)  
URL: <http://diver.stanford.edu/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Authoring Environment, Online Environment, Desktop Environment, Research Prototype, Annotation Tool  
Patterns: Annotations, Comments, Temporal Tags, Basic Controls, Direct Authoring  
Funktionalitätsrate: 0.1
  - **Dailymotion**  
Provider: Dailymotion  
URL: <http://www.dailymotion.com/de> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment  
Patterns: Search, Video Manager, Add Video, Related Videos, Sequential Media, Full Screen, Closed Captions, Viewing History, Loading Indicator, Multi-Timeline Editing, Basic Controls  
Funktionalitätsrate: 0.22
  - **EduBreak**  
Provider: Ghostthinker - Agentur für Wissensinnovationen  
URL: <http://www.ghostthinker.de/de/produkte/edubreak-campus> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Add Video, Annotations, Classified Marks, Annotated Timeline, Visual Highlighting, Comments, Overlays, Direct Authoring, Visual Summary, Search, Video Manager, Appropriate Delivery, Basic Controls  
Funktionalitätsrate: 0.26
  - **Electures-Portal**  
Provider: Albert-Ludwigs University Freiburg  
URL: <http://electures.informatik.uni-freiburg.de/portal/web/guest/home> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment

- Patterns: Search, Video Manager, Basic Controls  
Funktionalitätsrate: 0.06
- **Flowplayer**  
Provider: Flowplayer Ltd.  
URL: <http://www.flowplayer.org/> (abgerufen am 15.01.2016)  
Gruppierung: Video Framework, Online Environment, Open Source  
Patterns: Playlist, Closed Captions, Sequential Media, Hyperlinks, Overlays, User Traces, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.16
  - **Hyper-Hitchcock**  
Provider: Texas A&M University & FX Palo Alto Laboratory  
URL: (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Desktop Environment, Research Prototype  
Patterns: Add Video, Sequential Media, Annotated Timeline, Detail on Demand, Visual Summary, Basic Controls  
Funktionalitätsrate: 0.12
  - **HyperCafe**  
Provider: MIT MediaLab  
URL: (abgerufen am 15.01.2016)  
Gruppierung: Interactive Film, Research Prototype  
Patterns: Hyperlinks, Loop, Basic Controls  
Funktionalitätsrate: 0.06
  - **HyperSoap**  
Provider: MIT Media Labratory  
URL: <http://www.media.mit.edu/hypersoap/> (abgerufen am 15.01.2016)  
Gruppierung: Interactive Film, Research Prototype  
Patterns: Overlays, Object Tracking, Basic Controls  
Funktionalitätsrate: 0.06
  - **HyperVideo**  
Provider: Computer Graphics Center/Darmstadt and Knowledge Media Research Center/Tübingen  
URL: (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Research Prototype, Annotation Tool  
Patterns: Comments, Visual Highlighting, Annotations, Hyperlinks, Basic Controls  
Funktionalitätsrate: 0.1
  - **Hyperaudio Pad**  
Provider: Happyworm Ltd.  
URL: <http://happyworm.com> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Annotation Tool, Open Source  
Patterns: Add Video, Transcript, Video Manipulation, Media Fragments, Remix, Basic Controls  
Funktionalitätsrate: 0.12
  - **Hyperfilm**  
Provider: Hyperfilm Srl  
URL: <http://hyperfilm.newcloudsite.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Add Video, Hyperlinks, Simultaneous Media, Overlays, Detail on Demand, Assessment, Annotated Timeline, Annotations, Basic Controls  
Funktionalitätsrate: 0.18

– **IWRM education**

Provider: IHP/HWRP Secretariat, Federal Institute of Hydrology, Germany

URL: <http://www.iwrm-education.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Online Environment, Desktop Environment, Research Prototype, Open Source

Patterns: Hyperlinks, Simultaneous Media, Video Manager, Search, Detail on Demand, Table of Content, Loading Indicator, Annotated Timeline, Related Videos, Playlist, Classified Marks, Related Videos, Appropriate Delivery, Basic Controls

Funktionalitätsrate: 0.28

– **Interactive Dinner At Julia's**

Provider: MIT Media Laboratory

URL: (abgerufen am 15.01.2016)

Gruppierung: Interactive Film, Research Prototype

Patterns: Journaled Navigation, Object Tracking, Detail on Demand, Basic Controls

Funktionalitätsrate: 0.08

– **Internet Archive**

Provider: Internet Archive

URL: <https://archive.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment, Search Engine

Patterns: Related Videos, Visual Summary, Playlist, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.1

– **JoVE**

Provider: MyJoVE Corp.

URL: <http://www.jove.com> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment, Video Learning Environment

Patterns: Related Videos, Table of Content, Simultaneous Media, Transcript, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.12

– **Kaltura**

Provider: Kaltura Inc.

URL: <http://corp.kaltura.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Video Portal, Open Source

Patterns: Add Video, Table of Content, Search, Simultaneous Media, Closed Captions, Appropriate Delivery, Playlist, Loading Indicator, Remix, Multi-Timeline Editing, Annotated Timeline, User Traces, Related Videos, Basic Controls

Funktionalitätsrate: 0.28

– **Khan Academy**

Provider: Khan Academy

URL: <https://www.khanacademy.org> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Transcript, Assessment, Playback Speed, Basic Controls

Funktionalitätsrate: 0.08

– **Leanbackplayer**

Provider: Kapelan Media GmbH

URL: <http://www.leanbackplayer.com/> (abgerufen am 15.01.2016)

Gruppierung: Player, Online Environment

Patterns: Shortcut Commands, Basic Controls

Funktionalitätsrate: 0.04

### – **Lecture2Go**

Provider: Medienkompetenzzentrum, Regionales Rechenzentrum der Universität Hamburg

URL: <http://lecture2go.uni-hamburg.de/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment, Open Source

Patterns: Temporal Bookmarks, Follow Revisions, Appropriate Delivery, Media Fragments, Basic Controls

Funktionalitätsrate: 0.1

### – **Lecturio**

Provider: Lecturio GmbH

URL: <http://www.lecturio.de/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Video Manager, Search, User Ratings, Appropriate Delivery, Full Screen, Playback Speed, Sequential Media, Skip Back, Annotated Timeline, Table of Content, Loading Indicator, Related Videos, Basic Controls

Funktionalitätsrate: 0.26

### – **Lecturnity**

Provider: imc AG

URL: <http://www.lecturnity.de/de/lecturnity/uebersicht/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Desktop Environment, Annotation Tool, Video Learning Environment

Patterns: Visual Highlighting, Annotations, Table of Content, Overlays, Assessment, Video Manipulation, Assessment, Appropriate Delivery, Search, Multi-Timeline Editing, Shortcut Commands, Polls, Basic Controls

Funktionalitätsrate: 0.26

### – **MIT Video**

Provider: Massachusetts Institute of Technology

URL: <http://video.mit.edu/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment

Patterns: Search, Video Manager, Basic Controls

Funktionalitätsrate: 0.06

### – **MOVieGoer**

Provider: Zentrum für Graphische Datenverarbeitung (ZGDV) in Darmstadt

URL: <http://www.anjawipfler.de/hypervideo.html> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Research Prototype, Online Environment, Video Learning Environment

Patterns: Annotations, Overlays, Object Tracking, Basic Controls

Funktionalitätsrate: 0.08

### – **Magma Media Server**

Provider: Bildungsportal Sachsen GmbH

URL: <https://bildungsportal.sachsen.de/magma> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Video Authoring Environment, Online Environment

Patterns: Table of Content, Video Manager, Simultaneous Media, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.1

### – **Matterhorn**

Provider: ETH Zürich

URL: <http://demo.opencast.org:8080/welcome.html> (abgerufen am 15.01.2016)

Gruppierung: Open Source, Video Learning Environment, Online Environment, Video Portal, Player, Video Authoring Environment

Patterns: Video Manager, Classified Marks, Visual Summary, Transcript, Skip Back, Simultaneous Media, Annotations, Direct Authoring, Comments, Table of Content, User Traces, Search, Temporal Bookmarks, Shortcut Commands, Annotated Timeline, Basic Controls

Funktionalitätsrate: 0.32

– **MediaLoom**

Provider: John Tolva (Georgia Institute of Technology)

URL: <http://www.ascentstage.com/medialoom/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Desktop Environment, Research Prototype

Patterns: Add Video, Playback Speed, Simultaneous Media, Hyperlinks, Basic Controls

Funktionalitätsrate: 0.1

– **MediaSilo**

Provider: MediaSilo, Inc.

URL: <http://www.mediasilo.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Online Environment

Patterns: Add Video, Comments, Video Manager, Transcript, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.12

– **MediaSite**

Provider: Sonic Foundry, Inc.

URL: <http://www.sonicfoundry.com/mediasite> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Authoring Environment, Online Environment

Patterns: User Traces, Simultaneous Media, Comments, Playback Speed, Video Manager, Search, Temporal Bookmarks, Skip Back, Zoom, Related Videos, Basic Controls

Funktionalitätsrate: 0.22

– **MetaVid**

Provider: University of California Santa Cruz and the Sunlight Foundation.

URL: <http://metavid.ucsc.edu/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment, Open Source

Patterns: Temporal Bookmarks, Table of Content, Media Fragments, Transcript, Basic Controls

Funktionalitätsrate: 0.1

– **My Video**

Provider: MyVideo Broadband S.R.L.

URL: <http://www.myvideo.de/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment

Patterns: Add Video, Search, Playlist, User Ratings, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.12

– **NPTEL**

Provider: NPTEL

URL: <http://nptel.ac.in/courses.php?disciplineId=106> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Table of Content, Basic Controls

Funktionalitätsrate: 0.04

– **Ninsuna**

Provider: Multimedia Lab - Ghent University

URL: <http://ninsuna.elis.ugent.be/radiohead> (abgerufen am 15.01.2016)

Gruppierung: Interactive Film, Online Environment, Desktop Environment, Research Prototype

Patterns: Annotated Timeline, Overlays, Media Fragments, Hyperlinks, Basic Controls

Funktionalitätsrate: 0.1



### – **Non-linear Video**

Provider: Fraunhofer-Instituts für Offene Kommunikationssysteme FOKUS

URL: <http://www.fraunhofer.de/de/presse/filme/non-linear-video.html> (abgerufen am 15.01.2016)

Gruppierung: Online Environment, Research Prototype, Annotation Tool, Video Authoring Environment

Patterns: Overlays, Detail on Demand, Hyperlinks, Basic Controls

Funktionalitätsrate: 0.08

### – **Online Vorlesung**

Provider: Ulrich Bannier (Hamburg/DE)

URL: <http://www.online-vorlesungen.de/> (abgerufen am 15.01.2016)

Gruppierung: Misc, Video Portal

Patterns: Search, Basic Controls

Funktionalitätsrate: 0.04

### – **Open Video Annotation Project**

Provider: Center for Hellenic Studies, Harvard University & Junta de Andalucia

URL: <http://www.openvideoannotation.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Research Prototype, Annotation Tool, Open Source

Patterns: User Traces, Annotations, Overlays, Annotated Timeline, Loading Indicator, Basic Controls, Direct Authoring

Funktionalitätsrate: 0.14

### – **Open Video Player**

Provider: Open Video Player

URL: <http://openvideoplayer.sourceforge.net/html5video/> (abgerufen am 15.01.2016)

Gruppierung: Video Framework, Online Environment, Open Source

Patterns: Playlist, Sequential Media, Basic Controls

Funktionalitätsrate: 0.06

### – **OpenImages**

Provider: Netherlands Institute for Sound and Vision

URL: <http://www.openimages.eu/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment

Patterns: Related Videos, Search, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.08

### – **Opencast**

Provider: Universität Osnabrück/virtuOS and others

URL: <http://www.opencast.org> (abgerufen am 15.01.2016)

Gruppierung: Open Source, Video Learning Environment, Online Environment, Video Portal, Player, Video Authoring Environment

Patterns: Zoom, Annotated Timeline, Appropriate Delivery, Playback Speed, Table of Content, Transcript, Simultaneous Media, Shortcut Commands, Basic Controls

Funktionalitätsrate: 0.18

### – **PHP Melody**

Provider: PHPSUGAR

URL: <http://www.php sugar.com/phpmelody.html> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment

Patterns: Add Video, Follow Revisions, Related Videos, Loading Indicator, User Ratings, Video Manager, Search, Basic Controls

Funktionalitätsrate: 0.16

– **Paella Player**

Provider: Universitat Politecnica Valencia

URL: <http://paellaplayer.upv.es> (abgerufen am 15.01.2016)

Gruppierung: Online Environment, Open Source, Player

Patterns: Visual Summary, Appropriate Delivery, Closed Captions, Transcript, Table of Content, Search, Simultaneous Media, Playback Speed, Skip Back, Full Screen, Shortcut Commands, Basic Controls

Funktionalitätsrate: 0.24

– **Panopto**

Provider: Panopto

URL: <http://www.panopto.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment

Patterns: Add Video, Appropriate Delivery, Temporal Bookmarks, User Notes, Table of Content, Simultaneous Media, Search, User Ratings, User Traces, User Notes, Basic Controls

Funktionalitätsrate: 0.22

– **Playr**

Provider: Julien Villetorte

URL: <http://www.delphiki.com/html5/playr/> (abgerufen am 15.01.2016)

Gruppierung: Open Source, Video Framework, Player

Patterns: Closed Captions, Shortcut Commands, Basic Controls

Funktionalitätsrate: 0.06

– **Plumi**

Provider: EngageMedia

URL: <http://blog.plumi.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment, Open Source

Patterns: Add Video, Playlist, Related Videos, Video Manager, Closed Captions, Search, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.16

– **Popcorn.js & PopcornMaker**

Provider: Mozilla Foundation

URL: <http://popcornjs.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Video Authoring Environment, Video Framework, Online Environment, Open Source, Annotation Tool

Patterns: Add Video, Simultaneous Media, Hyperlinks, Overlays, Closed Captions, Loop, Synchronized Map, Remix, Annotations, Multi-Timeline Editing, Basic Controls, Break

Funktionalitätsrate: 0.24

– **Projekktor**

Provider: Sascha Kluer

URL: <http://www.projekktor.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Framework, Player

Patterns: Appropriate Delivery, Basic Controls, Annotated Timeline, Shortcut Commands

Funktionalitätsrate: 0.08

– **SIVA Producer**

Provider: University of Passau

URL: <http://www.mirkul.uni-passau.de/vorarbeiten/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Research Prototype, Annotation Tool, Desktop Environment

Patterns: Table of Content, Simultaneous Media, Hyperlinks, Detail on Demand, Basic Controls

Funktionalitätsrate: 0.1

### – **ScienceMovies**

Provider: VolkswagenStiftung

URL: <http://www.sciencemovies.de/en> (abgerufen am 15.01.2016)

Gruppierung: Online Environment, Video Portal

Patterns: Sequential Media, User Ratings, Loading Indicator, Video Manager, Basic Controls

Funktionalitätsrate: 0.1

### – **ShmoopTube**

Provider: Shmoop University

URL: <http://www.shmoop.com/video/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Related Videos, Transcript, Sequential Media, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.1

### – **Sofatutor**

Provider: sofatutor GmbH

URL: <http://www.sofatutor.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Related Videos, Assessment, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.08

### – **SproutVideo**

Provider: SproutVideo

URL: <https://sproutvideo.com> (abgerufen am 15.01.2016)

Gruppierung: Video Portal

Patterns: Playlist, User Traces, Add Video, Video Manager, Basic Controls

Funktionalitätsrate: 0.1

### – **Synote Media Fragment Player**

Provider: University of Southampton

URL: <http://smfplayer.synote.org/smfplayer/> (abgerufen am 15.01.2016)

Gruppierung: Video Framework, Online Environment, Research Prototype, Open Source, Player

Patterns: Media Fragments, Basic Controls

Funktionalitätsrate: 0.04

### – **TEDED**

Provider: TED

URL: <http://ed.ted.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment

Patterns: Add Video, Sequential Media, Assessment, Remix, Basic Controls

Funktionalitätsrate: 0.1

### – **The Mad Video**

Provider: The Mad Video Inc.

URL: <http://www.themadvideo.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment

Patterns: Add Video, Temporal Tags, Hyperlinks, Overlays, Annotations, Basic Controls

Funktionalitätsrate: 0.12

### – **Theresienstadt Explained**

Provider: nise81.com

URL: <http://theresienstadt-film.net> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment

Patterns: Synchronized Map, Visual Highlighting, Overlays, Simultaneous Media, Full Screen, Classified Marks, Video Manager, Multi-Timeline Editing, Annotations, Skip Back, Closed Captions,

Transcript, Annotated Timeline, Visual Summary, Zoom, Playback Speed, Temporal Tags, Table of Content, Shortcut Commands, Loading Indicator, Appropriate Delivery, Multi-Timeline Editing, Basic Controls

Funktionalitätsrate: 0.46

– **Timesheets.js**

Provider: Fabien Cazenave, INRIA/FR

URL: <http://wam.inrialpes.fr/timesheets/> (abgerufen am 15.01.2016)

Gruppierung: Video Framework, Online Environment, Open Source, Player

Patterns: Table of Content, Overlays, Simultaneous Media, Annotated Timeline, Basic Controls

Funktionalitätsrate: 0.1

– **Udacity**

Provider: Udacity, Inc

URL: <https://www.udacity.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Classified Marks, Annotated Timeline, Viewing History, Table of Content, Assessment, Sequential Media, User Traces, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.18

– **Unterrichtsmitschau 2.0**

Provider: LMU Munich

URL: <http://videoonline.edu.lmu.de/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment

Patterns: Search, Video Manager, Sequential Media, Visual Summary, Simultaneous Media, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.14

– **Viddix**

Provider: VIDDIX B.V

URL: <http://www.viddix.com/watch/4704edf4d1479/demo-viddix> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment

Patterns: Overlays, Assessment, Simultaneous Media, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.1

– **Video Lecture Jukebox**

Provider: Modul University Vienna

URL: <http://www.mediamixer.eu/jukebox/> (abgerufen am 15.01.2016)

Gruppierung: Online Environment, Research Prototype

Patterns: Sequential Media, Playlist, Media Fragments, Temporal Bookmarks, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.12

– **Video-Wiki**

Provider: Blankenship, Eric & Bakhtiar, Mikhak

URL: <http://dl.acm.org/citation.cfm?id=1297320> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Research Prototype

Patterns: Search, Transcript, Video Manipulation, Remix, Basic Controls

Funktionalitätsrate: 0.1

– **VideoAnt**

Provider: Academic Technology Services, College of Education & Human Development, University of Minnesota

URL: <https://ant2.cehd.umn.edu/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Research Prototype, Annotation

- Tool  
Patterns: Add Video, Comments, Annotations, Annotated Timeline, Basic Controls, Direct Authoring  
Funktionalitätsrate: 0.12
- **VideoClix**  
Provider: VideoClix  
URL: <http://www.videoclix.tv/> (abgerufen am 15.01.2016)  
Gruppierung: Online Environment  
Patterns: Object Tracking, Simultaneous Media, Overlays, Hyperlinks, Table of Content, User Traces, Annotations, Basic Controls  
Funktionalitätsrate: 0.16
  - **VideoLectures.NET**  
Provider: VideoLectures.NET  
URL: <http://videolectures.net/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Online Environment  
Patterns: Related Videos, Viewing History, Search, User Ratings, Table of Content, Annotated Timeline, Simultaneous Media, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.18
  - **VideoReadr**  
Provider: aflatune, Inc  
URL: <http://www.videoreadr.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Add Video, Transcript, Simultaneous Media, Basic Controls  
Funktionalitätsrate: 0.08
  - **VideoSolutions**  
Provider: Pearson Education, Inc.  
URL: <http://www.pearsonhighered.com/videosolutions/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Online Environment  
Patterns: Closed Captions, Basic Controls  
Funktionalitätsrate: 0.04
  - **VideoStir**  
Provider: VideoStir Ltd  
URL: <http://videostir.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment  
Patterns: Add Video, Simultaneous Media, Basic Controls  
Funktionalitätsrate: 0.06
  - **Videogram**  
Provider: Cinemacraft  
URL: <http://www.videogram.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment, Search Engine  
Patterns: Video Manager, Visual Summary, Related Videos, Search, Basic Controls  
Funktionalitätsrate: 0.1
  - **Vimeo**  
Provider: Vimeo, LLC  
URL: <http://vimeo.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment, Search Engine  
Patterns: Add Video, Search, Video Manager, Playlist, Full Screen, Sequential Media, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.16

– **WhiteBoardPlayer**

Provider: Albert-Ludwigs University Freiburg

URL: (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment

Patterns: Comments, Classified Marks, Annotations, Simultaneous Media, Table of Content, Search, Overlays, Basic Controls

Funktionalitätsrate: 0.16

– **Wiki Commons**

Provider: Wikimedia Foundation

URL: <http://commons.wikimedia.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment, Search Engine, Open Source

Patterns: Add Video, Sequential Media, Appropriate Delivery, Video Manager, Search, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.14

– **Wistia**

Provider: WISTIA

URL: <http://wistia.com> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment

Patterns: User Traces, Appropriate Delivery, Comments, Basic Controls

Funktionalitätsrate: 0.08

– **XIMPEL**

Provider: XIMPEL

URL: <http://www.ximpel.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Authoring Environment, Online Environment, Research Prototype, Open Source, Annotation Tool

Patterns: Add Video, Playlist, Overlays, Hyperlinks, Viewing History, Assessment, Playlist, Basic Controls

Funktionalitätsrate: 0.16

– **Yendif Player**

Provider: Yendif Player

URL: <http://yendifplayer.com> (abgerufen am 15.01.2016)

Gruppierung: Player, Online Environment

Patterns: Shortcut Commands, Playlist, Closed Captions, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.1

– **YoVisto**

Provider: yovisto GmbH

URL: <http://www.yovisto.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Annotated Timeline, Visual Summary, Simultaneous Media, User Traces, Temporal Tags, Comments, Basic Controls

Funktionalitätsrate: 0.14

– **YouKu**

Provider: Youku Tudou Inc.

URL: <http://www.youku.com> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment

Patterns: Add Video, Viewing History, Video Manager, Search, Related Videos, Basic Controls

Funktionalitätsrate: 0.12

– **YouTube**

Provider: YouTube Inc.

URL: <http://www.youtube.com/edu> (abgerufen am 15.01.2016)

Gruppierung: Video Portal, Online Environment, Search Engine, Video Authoring Environment, Annotation Tool

Patterns: Related Videos, Closed Captions, Annotations, Add Video, Journaled Navigation, Search, Video Manager, Loading Indicator, User Traces, Appropriate Delivery, Viewing History, Playlist, Branching Videos, Visual Highlighting, Shortcut Commands, User Ratings, Sequential Media, Hyperlinks, Overlays, Visual Summary, Multi-Timeline Editing, Basic Controls

Funktionalitätsrate: 0.44

### – **Zeugen der Shoa**

Provider: CDIS, FU Berlin

URL: <http://www.zeugendershoah.de/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment, Search Engine

Patterns: Sequential Media, Transcript, Search, Video Manager, Basic Controls

Funktionalitätsrate: 0.1

### – **cacophonyjs**

Provider: Johnny Broadway

URL: <http://www.cacophonyjs.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Framework, Online Environment, Open Source, Player

Patterns: Branching Videos, Inline Drawing, Assessment, Basic Controls

Funktionalitätsrate: 0.08

### – **coursera**

Provider: Coursera Inc.

URL: <https://www.coursera.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment

Patterns: Playback Speed, Sequential Media, Closed Captions, Video Manager, Loading Indicator, Basic Controls

Funktionalitätsrate: 0.12

### – **digital-tutors**

Provider: video-tutors

URL: <http://www.digitaltutors.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Online Environment, Annotation Tool

Patterns: Playlist, Video Manager, Sequential Media, Temporal Tags, User Notes, Search, Viewing History, User Notes, Journaled Navigation, Annotated Timeline, Classified Marks, Loading Indicator, Direct Authoring, Basic Controls

Funktionalitätsrate: 0.28

### – **edX**

Provider: edX Inc.

URL: <https://www.edx.org/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Video Framework, Online Environment, Desktop Environment, Research Prototype, Open Source

Patterns: Transcript, Closed Captions, Playback Speed, Sequential Media, Simultaneous Media, Full Screen, Basic Controls

Funktionalitätsrate: 0.14

### – **eduCanon**

Provider: eduCanon

URL: <https://www.educanon.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Authoring Environment, Online Environment

Patterns: Add Video, Assessment, Annotated Timeline, User Traces, Hyperlinks, Annotations, Vi-



- deo Manager, Basic Controls, Break  
Funktionalitätsrate: 0.18
- **huzzaz**  
Provider: Huzzaz  
URL: <https://huzzaz.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment  
Patterns: Add Video, Sequential Media, Playlist, Basic Controls  
Funktionalitätsrate: 0.08
  - **iTunes U**  
Provider: Apple Inc.  
URL: <http://www.apple.com/de/education/itunes-u/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Online Environment  
Patterns: Search, Video Manager, Basic Controls  
Funktionalitätsrate: 0.06
  - **iversity**  
Provider: iversity GmbH  
URL: <https://iversity.org/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment  
Patterns: Add Video, Appropriate Delivery, Playback Speed, Assessment, Video Manager, Basic Controls  
Funktionalitätsrate: 0.12
  - **ivotek**  
Provider: ivotek  
URL: <http://www.ivotek.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Object Tracking, Detail on Demand, Visual Highlighting, Overlays, Basic Controls  
Funktionalitätsrate: 0.1
  - **lernfunk.de**  
Provider: virtUOSZentrum für Informationsmanagement und virtuelle Lehre, Universität Osnabrück  
URL: <http://lernfunk.de/portal-ui/user-interface/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment, Open Source, Video Learning Environment  
Patterns: User Traces, Appropriate Delivery, Table of Content, Follow Revisions, Sequential Media, Shortcut Commands, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.16
  - **metta.io**  
Provider: Dragontape Ltd./HU  
URL: <http://www.metta.io/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Classified Marks, Add Video, Assessment, Polls, Annotated Timeline, Simultaneous Media, Overlays, Hyperlinks, Comments, Loading Indicator, Basic Controls, Annotations  
Funktionalitätsrate: 0.24
  - **openHPI**  
Provider: Hasso Plattner Institute Potsdam  
URL: <https://open.hpi.de/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment, Research Prototype

- Patterns: Playback Speed, Simultaneous Media, User Traces, Sequential Media, Basic Controls  
Funktionalitätsrate: 0.1
- **openhypervideo**  
Provider: Sascha Jäger, filmicweb  
URL: <http://www.open-hypervideo.org/> (abgerufen am 15.01.2016)  
Gruppierung: Online Environment, Research Prototype  
Patterns: Add Video, Hyperlinks, Sequential Media, Simultaneous Media, Annotated Timeline, Basic Controls  
Funktionalitätsrate: 0.12
  - **podcampus**  
Provider: Multimedia Kontor Hamburg GmbH  
URL: <http://www.podcampus.de/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Online Environment, Desktop Environment, Search Engine  
Patterns: Search, Video Manager, Basic Controls  
Funktionalitätsrate: 0.06
  - **rivafx**  
Provider: memex GmbH  
URL: <http://rivavx.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment  
Patterns: Hyperlinks, Detail on Demand, Basic Controls  
Funktionalitätsrate: 0.06
  - **tele-TASK**  
Provider: Hasso Plattner Institute  
URL: <http://www.tele-task.de/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Online Environment  
Patterns: Annotated Timeline, Simultaneous Media, User Ratings, Table of Content, Annotations, Playlist, Related Videos, Search, Video Manager, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.22
  - **touchcast**  
Provider: touchcast  
URL: <http://www.touchcast.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment, Annotation Tool  
Patterns: Classified Marks, Add Video, Synchronized Map, Detail on Demand, Hyperlinks, Annotated Timeline, Annotations, Branching Videos, Overlays, Loading Indicator, Related Videos, Video Manipulation, Basic Controls  
Funktionalitätsrate: 0.26
  - **transLectures**  
Provider: transLectures.NET  
URL: <https://www.translectures.eu/> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Authoring Environment, Online Environment, Research Prototype  
Patterns: Add Video, Table of Content, Transcript, Simultaneous Media, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.12
  - **veotag**  
Provider: Veotag.com  
URL: (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Authoring Environment, Online Environment, Annotation Tool

- Patterns: Table of Content, Comments, Search, Basic Controls  
Funktionalitätsrate: 0.08
- **vi-wiki**  
Provider: nise81.com  
URL: <https://p2pu.org/en/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Research Prototype, Annotation Tool, Open Source  
Patterns: Add Video, Hyperlinks, Overlays, Annotations, Loading Indicator, Basic Controls, Direct Authoring  
Funktionalitätsrate: 0.14
  - **vialogues**  
Provider: EdLab, Teachers College, Columbia University  
URL: <https://vialogues.com/> (abgerufen am 15.01.2016)  
Gruppierung: Online Environment, Video Portal  
Patterns: Add Video, Comments, Annotations, User Traces, Related Videos, Direct Authoring, Basic Controls  
Funktionalitätsrate: 0.14
  - **videoNot.es**  
Provider: VideoNot.es  
URL: <http://www.videonot.es/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Annotation Tool, Open Source  
Patterns: User Notes, Annotations, Table of Content, Transcript, Direct Authoring, Basic Controls  
Funktionalitätsrate: 0.12
  - **virtpresenter**  
Provider: virtUOS - Center for Information Management and Virtual Teaching, University of Osnabrück  
URL: <http://www.virtuos.uni-osnabrueck.de/VirtPresenter/Beispiele> (abgerufen am 15.01.2016)  
Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment, Desktop Environment, Research Prototype  
Patterns: Simultaneous Media, User Traces, Table of Content, Annotated Timeline, Basic Controls  
Funktionalitätsrate: 0.1
  - **vzaaar**  
Provider: vzaar Limited  
URL: <http://vzaar.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Video Framework, Online Environment, Video Authoring Environment  
Patterns: Loop, Temporal Bookmarks, Journaled Navigation, Table of Content, Sequential Media, Add Video, Video Manager, Overlays, Loading Indicator, Basic Controls  
Funktionalitätsrate: 0.2
  - **webcasts.berkeley**  
Provider: UC Berkeley  
URL: <http://webcast.berkeley.edu/> (abgerufen am 15.01.2016)  
Gruppierung: Video Portal, Online Environment  
Patterns: Video Manager, Basic Controls  
Funktionalitätsrate: 0.04
  - **wireWAX**  
Provider: wireWAX Inc.  
URL: <https://www.wirewax.com/> (abgerufen am 15.01.2016)  
Gruppierung: Video Authoring Environment, Online Environment, Misc  
Patterns: Annotated Timeline, Object Tracking, Overlays, Detail on Demand, Classified Marks,

## A.5. VERZEICHNIS DER UNTERSUCHTEN VIDEOSYSTEME

Loading Indicator, Basic Controls

Funktionalitätsrate: 0.14

– **zaption**

Provider: Zaption Inc.

URL: <http://www.zaption.com/> (abgerufen am 15.01.2016)

Gruppierung: Video Learning Environment, Video Portal, Video Authoring Environment, Online Environment

Patterns: Add Video, Annotated Timeline, Assessment, Sequential Media, Annotated Timeline, User Traces, Simultaneous Media, Basic Controls

Funktionalitätsrate: 0.16

## Kodierschema der betrachteten Videosysteme

- Name / Bezeichnung der Anwendung
- Anbieter (Personen-, Firmen- oder Organisationsbezeichnung)
- URL der Anwendung
- Gruppe der Anwendung (mehrfache Auswahl möglich):
  - Videolernumgebung
  - Portal
  - Autorenumgebung
  - Video-Framework
  - Interaktiver Film
  - Online-Anwendung
  - Desktop-Anwendung
  - Forschungsprototyp
  - Annotationswerkzeug
  - Video Player
- Auflistung der identifizierten Kategorien
- Beschreibung der Anwendung einschließlich der Merkmalsausprägung je identifizierter Kategorie.
- Angaben zur Verfügbarkeit, z.B.
  - verfügbar / nicht mehr verfügbar seit / zu letzt verfügbar gewesen am . . . ,
  - online / offline,
  - mit oder ohne Registrierung,
  - Zugang auf Anfrage eingeräumt,
  - Demo-Zugang / beschränkter Zugang / Premium-Zugang,
  - alternative Beschreibungen in Publikationen oder Produktdokumentation.
- Persönlich angewendet: ja / nein

## Interrater-Reliabilität

Zur Sicherstellung der Intersubjektivität des Categoriesystems führte eine zweite Person eine unabhängige Kategorisierung durch. Bei dem Bewerter handelte es sich um Avid Segal (Ben Gurion Universität), der die Bewertung im Zeitraum vom März bis Juli 2015 durchführte. In der Tab. A.10 sind die Übereinstimmungen und Unterschiede beider Bewerter dargestellt. Zur Bestimmung der Interrater-Reliabilität wurde Cohens Kappa ( $\kappa$ ) je betrachteten Informationsträger und je Analysekategorie ermittelt:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (\text{A.2})$$

**Tabelle A.10:** Ergebnisse des zweiten Schätzers. Dargestellt sind übereinstimmende Zuordnungen (●) und Nicht-Zuordnungen (○) einer Kategorie sowie die vom zweiten Schätzer zusätzlich kodierten Kategorien (+) und die von ihm nicht kodierten Kategorien (–).

Merkmalskategorien	Amazonie	HyperCafe	Hyperfilm	Khan Academy	Lecture2Go	Lecturio	lernfunk.de	MediaSite	MIT Video	Opencast	PHP Melody	Plumi	Timesheets.js	transLectures	Udacity	Video-Wiki	VideoAnt	VideoLectures.NET	videoNotes	Vimeo	Yendif Player	YouTube	YoVisto	zaption	Zeugen der Shoa	$\kappa$
BASIC CONTROLS	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	1,00
APPROPRIATE DELIVERY	○	○	○	–	●	●	●	○	○	●	○	○	○	○	–	○	○	○	○	–	○	●	○	○	○	0,86
LOADING INDICATOR	●	○	○	–	○	●	●	○	○	○	●	●	○	●	●	○	○	●	○	●	●	●	○	○	○	0,95
SHORTCUT COMMANDS	–	○	○	○	–	○	●	○	○	●	○	○	○	○	–	○	○	○	–	–	●	●	○	–	○	0,72
SEARCH	○	○	○	○	○	●	○	●	●	○	●	●	○	○	○	●	○	●	○	●	○	●	○	○	●	1,00
TABLE OF CONTENT	○	○	○	○	○	●	●	○	○	●	○	○	●	●	●	○	○	+	●	○	○	○	○	○	○	0,95
TEMPORAL TAGS	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	1,00
TEMPORAL BOOKMARKS	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1,00
PLAYBACK SPEED	○	○	○	●	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1,00
PLAYBACK DIRECTION	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1,00
ZOOM	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1,00
VISUAL SUMMARY	○	○	○	○	○	○	–	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	0,96
ANNOTATED TIMELINE	●	○	●	○	○	●	○	○	○	●	○	○	●	○	●	○	●	●	○	○	○	○	○	●	●	1,00
TRANSCRIPT	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	●	○	○	+	○	○	○	○	○	○	0,95
CLOSED CAPTIONS	●	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	●	○	○	○	1,00
SKIP BACK	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1,00
JOURNALED NAVIGATION	○	○	○	○	–	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	0,96
LOOP	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1,00
ADD VIDEO	○	○	●	○	○	○	○	○	○	○	●	●	○	●	○	○	●	○	–	●	○	●	–	●	○	0,90
ANNOTATIONS	○	○	●	○	○	○	○	○	○	○	○	○	–	○	○	○	●	○	●	○	○	●	○	○	○	0,95
COMMENTS	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	1,00

(Fortsetzung auf der nächsten Seite)

(Tabelle A.10 – Fortsetzung von der vorherigen Seite)

Merkmalskategorien	Amazonie	HyperCafe	Hyperfilm	Khan Academy	Lecture2Go	Lecturio	lernfunk.de	MediaSite	MIT Video	Opencast	PHP Melody	Plumi	Timesheets.js	transLectures	Udacity	Video-Wiki	VideoAnt	VideoLectures.NET	videoNotes	Vimeo	Yendif Player	YouTube	YoVisto	zaption	Zeugen der Shoa	$\kappa$
INLINE DRAWING	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
POLLS	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
DIRECT AUTHORING	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	●	o	o	o	o	–	o	0,96
REMIX	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	o	o	o	o	1,00
VIDEO MANIPULATION	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	+	o	o	o	0,96
MULTI-TIMELINE EDITING	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	1,00
VIDEO MANAGER	–	o	o	–	–	●	o	●	●	o	●	●	o	o	o	o	o	o	o	●	o	●	o	o	●	0,85
BREAK	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
SEQUENTIAL MEDIA	●	o	o	o	o	●	●	o	o	o	o	o	o	o	+	o	o	o	o	●	o	●	o	●	●	0,95
RELATED VIDEOS	●	o	o	o	o	●	o	●	o	o	●	●	o	o	o	o	o	●	o	o	o	●	o	o	o	1,00
HYPERLINKS	●	●	●	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	1,00
BRANCHING VIDEOS	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	1,00
DETAIL ON DEMAND	o	o	●	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	–	o	o	o	0,96
MEDIA FRAGMENTS	o	o	o	o	●	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
CLASSIFIED MARKS	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	o	o	o	o	o	1,00
USER RATINGS	o	o	o	o	o	●	o	o	o	o	●	o	o	o	o	o	o	●	o	–	o	●	o	o	o	0,95
VIEWING HISTORY	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	●	o	o	o	●	o	o	o	1,00
PLAYLIST	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	o	o	●	●	●	o	o	o	1,00
FOLLOW REVISIONS	o	o	o	o	+	o	●	o	o	o	●	o	o	o	o	o	o	o	o	o	o	o	o	o	o	0,96
USER NOTES	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	o	o	1,00
ASSESSMENT	o	o	●	+	o	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	o	o	o	●	o	0,95
VIDEO CLIP QUEST	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
USER TRACES	o	o	o	o	o	o	●	●	o	o	o	o	o	o	●	o	o	o	o	o	o	●	●	●	o	1,00
FULL SCREEN	o	o	o	o	o	●	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	–	o	o	o	0,96
SIMULTANEOUS MEDIA	o	o	●	o	o	o	o	●	o	●	o	o	●	●	o	o	o	●	o	o	o	o	●	●	o	1,00
SYNCHRONIZED MAP	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
OVERLAYS	●	o	●	o	o	o	o	o	o	o	o	o	●	o	o	o	o	o	o	o	o	●	o	o	o	1,00
VISUAL HIGHLIGHTING	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	●	o	o	o	1,00
OBJECT TRACKING	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00
MULTI-ANGLE VIDEO	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	1,00

(Fortsetzung auf der nächsten Seite)



(Tabelle A.10 – Fortsetzung von der vorherigen Seite)

Merkmalskategorien	
0,95 Amazonie	$\kappa$
1,00 HyperCafe	
1,00 Hyperfilm	
0,92 Khan Academy	
0,91 Lecture2Go	
1,00 Lecturio	
0,98 lernfunk.de	
1,00 MediaSite	
1,00 MIT Video	
1,00 Opencast	
1,00 PHP Melody	
1,00 Plumi	
0,98 Timesheets.js	
1,00 transLectures	
0,93 Udacity	
1,00 Video-Wiki	
1,00 VideoAnt	
0,98 VideoLectures.NET	
0,93 videoNotes	
0,93 Vimeo	
1,00 Yendif Player	
0,92 YouTube	
0,98 YoVisto	
0,96 zaption	
1,00 Zeugen der Shoa	
	$\kappa$

## Ergebnisse der Inhaltsanalyse

### Ergebnisse der je Gruppe von Informationsträgern

**Tabelle A.11:** Vergleich der Analysedimensionen bezogen auf die zugehörigen Kategorien. Häufigkeit, arithmetischen Mittels (*M*) und Standardabweichung (*SD*) bezogen auf die Kategorien und identifizierten Merkmalsausprägungen je Dimension.

	Kategorien in der Gruppe Learning Environment				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	4	73	22.53	18.25	14.25
Layout	6	33	10.19	5.50	5.85
Zugang zu zeitbasierter Information	12	92	28.40	7.67	5.37
Strukturierung	9	56	17.28	6.22	5.09
Selbstorganisation	6	30	9.26	5.00	3.61
Kontribution	8	40	12.35	5.00	3.77
Gesamt	45	324	100	7.20	7.27

	Kategorien in der Gruppe Authoring Environment				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	4	82	19.85	20.50	20.12
Kontribution	8	86	20.82	10.75	9.08
Layout	6	55	13.32	9.17	8.51
Zugang zu zeitbasierter Information	13	87	21.07	6.69	5.06
Strukturierung	10	68	16.46	6.80	4.64
Selbstorganisation	6	35	8.47	5.83	3.58
Gesamt	47	413	100	8.79	9.30

	Kategorien in der Gruppe Portal				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit/%	<i>M</i>	<i>SD</i>
Grundfunktionen	4	100	23.64	25.00	21.58
Strukturierung	10	91	21.51	9.10	7.97
Zugang zu zeitbasierter Information	13	100	23.64	7.69	6.56
Kontribution	8	54	12.77	6.75	6.67
Layout	6	33	7.80	5.50	5.32
Selbstorganisation	6	45	10.64	7.50	4.54
Gesamt	47	423	100	9.00	10.19

**Tabelle A.12:** Vergleich der Analysedimensionen bezogen auf die zugehörigen Kategorien. Häufigkeit, arithmetischen Mittels (*M*) und Standardabweichung (*SD*) bezogen auf die Kategorien und identifizierten Merkmalsausprägungen je Dimension.

	Kategorien in der Gruppe Framework				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	4	15	25.86	3.75	3.63
Strukturierung	6	10	17.24	1.67	1.11
Layout	4	9	15.52	2.25	1.30
Zugang zu zeitbasierter Information	8	14	24.14	1.75	0.97
Selbstorganisation	3	4	6.90	1.33	0.47
Kontribution	5	6	10.34	1.20	0.40
Gesamt	30	58	100	1.93	1.77

	Kategorien in der Gruppe Annotation Tool				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit/%	<i>M</i>	<i>SD</i>
Grundfunktionen	4	38	17.12	9.50	9.60
Kontribution	8	64	28.83	8.00	5.52
Layout	5	29	13.06	5.80	4.26
Strukturierung	10	36	16.22	3.60	3.35
Zugang zu zeitbasierter Information	9	36	16.22	4.00	2.83
Selbstorganisation	5	19	8.56	3.80	1.17
Gesamt	41	222	100	5.41	5.13

	Kategorien in der Gruppe Player				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	4	23	31.51	5.75	4.21
Zugang zu zeitbasierter Information	10	29	39.73	2.90	1.45
Layout	3	7	9.59	2.33	1.25
Kontribution	4	6	8.22	1.50	0.50
Selbstorganisation	3	4	5.48	1.33	0.47
Strukturierung	4	4	5.48	1.00	0.00
Gesamt	28	73	100	2.61	2.38

**Tabelle A.13:** Vergleich der Analysedimensionen bezogen auf die zugehörigen Kategorien. Häufigkeit, arithmetischen Mittels (*M*) und Standardabweichung (*SD*) bezogen auf die Kategorien und identifizierten Merkmalsausprägungen je Dimension.

	Kategorien in der Gruppe Research Prototype				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	3	31	22.14	10.33	10.50
Strukturierung	7	28	20.00	4.00	3.25
Layout	5	21	15.00	4.20	3.25
Zugang zu zeitbasierter Information	11	29	20.71	2.64	2.10
Kontribution	6	22	15.71	3.67	2.29
Selbstorganisation	4	9	6.43	2.25	1.30
Gesamt	36	140	100	3.89	4.40

	Kategorien in der Gruppe Open Source				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	4	46	23.47	11.50	8.44
Zugang zu zeitbasierter Information	11	51	26.02	4.64	3.02
Kontribution	8	32	16.33	4.00	2.50
Layout	5	18	9.18	3.60	2.87
Selbstorganisation	6	21	10.71	3.50	2.36
Strukturierung	9	28	14.29	3.11	1.59
Gesamt	43	196	100	4.56	4.20

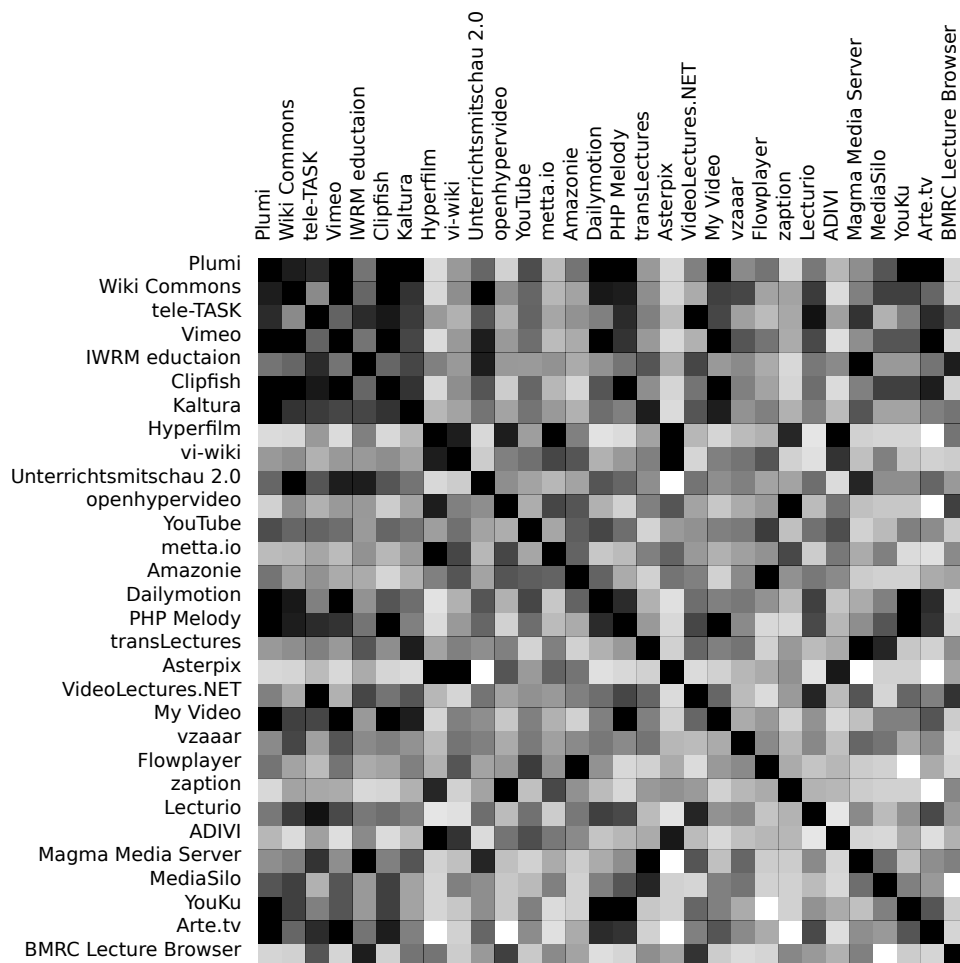
	Kategorien in der Gruppe Interactive Film				
	Anzahl	abs. Häufigkeit	rel. Häufigkeit	<i>M</i>	<i>SD</i>
Grundfunktionen	3	9	24.32	3.00	2.83
Layout	2	7	18.92	3.50	1.50
Strukturierung	7	14	37.84	2.00	1.41
Zugang zu zeitbasierter Information	5	7	18.92	1.40	0.80
Gesamt	17	37	100	2.18	1.79

## Ähnlichkeit der Informationsträger

Aus den Kategoriezuordnungen ergibt sich die Möglichkeit, die Ähnlichkeit der Informationsträger zu bestimmen. Zur Berechnung der Ähnlichkeit gibt es viele unterschiedliche Verfahren. Die Ähnlichkeit zweier Lernumgebungen  $i_a$  und  $i_b$  lässt sich beispielsweise durch den Jaccard-Index quantifizieren:

$$J(i_a, i_b) = \frac{|K_{i_a} \cap K_{i_b}|}{|K_{i_a} \cup K_{i_b}|} \quad (\text{A.3})$$

In der *Shaded Similarity Matrix* in Abb. A.2 sind die 40 Videolernumgebungen mit dem höchsten Jaccard-Index angeführt. Cluster entlang der Diagonalen deuten auf sehr ähnlicher Lernumgebungen hin.



**Abbildung A.2:** *Shaded Similarity Matrix* von 40 Videolernumgebungen, die den höchsten Jaccard-Index aufweisen. Schwarze Felder in dieser Heatmap kennzeichnen einen Jaccard-Index von 1, d.h. eine vollständige Übereinstimmung der Kategorisierung. Ist ein Feld hingegen weiß, so besitzen die Lernumgebungen keine übereinstimmenden Kategorien – der Jaccard-Index ist Null.

Der Jaccard-Index vernachlässigt jedoch die Ähnlichkeit zweier Systeme aufgrund von Kategorien, die ihnen beiden nicht zugeordnet werden konnten. Beispielsweise zeichnen sich Videoplayer dadurch aus, dass sie über keine Funktionalitäten respektive Kategorien im Sinne der Makrointeraktivität verfügen. Für einen Vergleich zweier Videolernumgebungen  $a$  und  $b$  ist eine Darstellung der Kategorien als binären Vektoren  $v_a$  und  $v_b$  hilfreich. Die Länge eines Vektors entspricht der Anzahl an Kategorien. Für jede kodierte Kategorie wird an der entsprechenden

Position im Vektor eine 1 vermerkt. Wurde eine Kategorie nicht vermerkt, wird eine 0 verzeichnet.

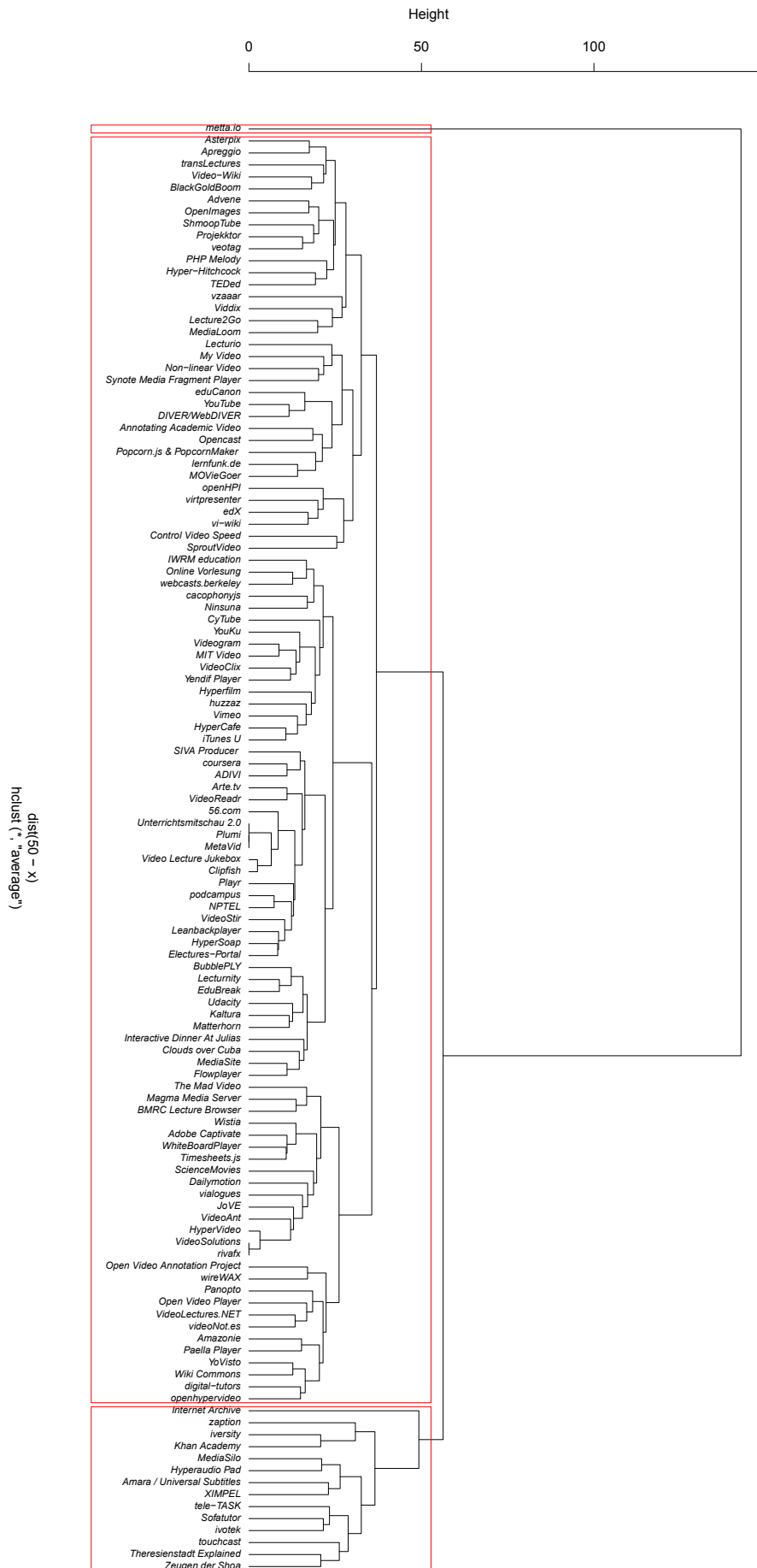
Um also auch nicht vorhandene Kategorien als Gemeinsamkeiten bei der Ermittlung der Ähnlichkeit zu berücksichtigen, kann man mit Hilfe dieser Vektoren den Hamming Abstand zweier Lernumgebungen  $a$  und  $b$  ermitteln. Dieser ergibt sich aus der Anzahl ungleicher Werte im binären Vektor, der aus der bitweisen XOR-Operation der beiden binären Vektoren  $v_a$  und  $v_b$  resultiert:

$$\Delta(i_a, i_b) := |\{j \in \{1, \dots, n\} \mid i_{a,j} \neq i_{b,j}\}| \quad (\text{A.4})$$

Im Dendrogramm in Abb. A.3 ist das Ergebnis einer hierarchischen Clusteranalyse (UPGMA) hinsichtlich der Kategorien, welche die Mikrointeraktivität der Informationsträger beschreiben, dargestellt. Der binäre Baum besteht dabei aus Informationsträgern, die zueinander eine geringe Hamming Distanz aufweisen. Vier Cluster ließen sich dabei identifizieren, wobei die letzten beiden aufgrund ihrer geringen Größe nicht relevant sind:

- Ambitionierte Videoplayer (28): Player, die über eine große Anzahl an Features verfügen.
- Gewöhnliche Videoplayer (90): Player, die kaum mehr als die grundlegenden Funktionen bieten.
- Matterhorn und Paella Player (2): Zwei alternative Player für das Vortragsaufzeichnungssystem Matterhorn/Openecast.
- Theresienstadt Explained (1): Anwendung, die aufgrund ihres sehr hohen Funktionsumfangs sehr viele Überschneidungen mit anderen Playern aufweist.

## A.8. ERGEBNISSE DER INHALTSANALYSE



**Abbildung A.3:** Dendrogramm der hierarchischen Clusterisierung der Informationsträger hinsichtlich der Anzahl gemeinsamer Kategorien.



# Ergebnisse der Analyse ähnlicher Muster

## Designing Interfaces: Patterns for effective Interaction Design

Source: [Tidwell 2005] and [Tidwell 2011]

Patterns: 125 (1st/2nd edition)

- KEYBOARD ONLY [Tidwell 2005, p. 17]  
→ Is refined by SHORTCUT COMMANDS.
- WIZZARD: „Lead the user through the interface step by step, doing tasks in a prescribed order“ [Tidwell 2005, p. 42-44]  
→ Is refined by SEQUENTIAL MEDIA.
- EXTRAS ON DEMAND: „Show the most important content upfront, but hide the rest. Let the user reach it via a single, simple gesture.“ [Tidwell 2005, p. 45-46]  
→ Is refined by DETAILS ON DEMAND
- ALTERNATIVE VIEWS „Let the user choose among alternative views that are substantially different from the default view.“ [Tidwell 2011, p. 64-67]  
→ Is refined by MULTI-ANGLE VIDEOS
- ANNOTATED SCROLLBAR: „Make the scrollbar serve double-duty as a map of content and a “You are here” indicator.“ [Tidwell 2005, p. 80-81]  
→ is refined by ANNOTATED TIMELINE
- PYRAMID: „Link a sequence of pages with NEXT/BACK links. Combine this sequential presentation with a main page that links to and from all pages in the sequence.“ [Tidwell 2005, p. 71-73]  
→ Is complemented by SEQUENTIAL MEDIA.
- SEQUENCE MAP: „On each page in a sequence, show a map of all of the pages in order, including a “You are here” indicator.“ [Tidwell 2005, p. 76-77]  
→ Is complemented by STORY GRAPH.
- CLOSABLE PANELS: „Put sections of content onto separate panels, and let the user open and close each of them separately from the other.“ [Tidwell 2005, p. 111-113]  
→ Is refined by OVERLAYS.
- PREVIEW: „Show users a preview or summary of what will happen if the perform an action.“ [Tidwell 2005, p. 147-148]  
→ Is refined by VISUAL SUMMARY.
- PROGRESS INDICATOR: „Show the user how much progress was made on a time-consuming operation.“ [Tidwell 2005, p. 149-150]  
→ Is almost identically with LOADING INDICATOR.
- COMMAND HISTORY: „As the user performs actions, keep a visible record of what was done, to what and when.“ [Tidwell 2005, p. 156-157]  
→ Is refined by FOLLOW REVISIONS.
- OVERVIEW PLUS DETAIL: „Place an overview of the graphic next to the zoomed “detail view”. As the user drags a viewport around the overview, show that part of the graphic in the detail view.“ [Tidwell 2005, p. 174-175]  
→ Is refined by ZOOM.

- INPUT PROMPT: „Prefill a text field or drop down with a prompt that tells the user what to do or type.“ [Tidwell 2005, p. 225-226]  
→ Is refined by DISCUSSION PROMPT.
- EDIT IN PLACE: „Use a small, dynamic text editor to let the user change text “in place”: position the editor directly over the original text, rather than using a separate panel or dialog box.“ [Tidwell 2005, p. 249-250]  
→ Is abstracted by DIRECT AUTHORING.
- PICTURE MANAGER: „Use thumbnails, item views, and a browsing interface to create a familiar structure for managing photos, videos, and other pictorial items.“ [Tidwell 2011, p. 40-44], see also <http://designinginterfaces.com/patterns/picture-manager/>  
→ Is refined by VIDEO MANAGER.
- LOADING INDICATORS [Tidwell 2011, p. 468-469]  
→ Is refined by LOADING INDICATOR.

### **User-Centered Interaction Design Patterns for Interactive Digital Television Applications**

Source: Kunert [2009]

Patterns: 42

- OVERLAY: „Overlay applications cover little of the screen, letting the main action in background video be seen well enough. Overlays are suitable for showing small amounts of texts or a particular context-sensitive function directly related to the program flow. Overlays are used mainly for enhanced TV applications, e.g. for enhancements of quiz shows, documentaries or sports programs [...]“ [Kunert 2009, p. 109]  
→ Is refined by OVERLAYS, CLOSED CAPTIONS, VISUAL HIGHLIGHTING, and ASSESSMENT.
- MENU: „A menu provides access to different content items and functions. Often a menu item leads to a sub-menu. Such a menu provides access to hierarchically organised contents and functions.“ [Kunert 2009, p. 114]  
→ Is refined by TABLE OF CONTENT and TEMPORAL TAGS.
- VIDEO MULTI-SCREEN: „A video multi-screen consists of several video streams shown at the same time. [...] In video multi-screen users can choose any of the presented video streams and enlarge it to full-screen size.“ [Kunert 2009, p. 117]  
→ Is refined by SIMULTANEOUS MEDIA and abstracted by VIDEO MANAGER. It is also almost identical with MULTI-ANGLE VIDEO.
- INDEX: „An index offers access to an alphabetically organised overview of the content and functions“ [Kunert 2009, p. 119]  
→ Is refined by TABLE OF CONTENT and TEMPORAL TAGS
- CHOOSING THE RIGHT KEYS: „The design of remote controls varies widely. Some keys are standardised and provided by every remote control: ARROW KEYS [...], OK KEY [...], COLOUR KEYS [...], NUMBER KEYS [...].“ [Kunert 2009, p. 122]  
→ Is refined by SHORTCUT COMMANDS.
- LOADING INDICATION: „Indicate the application status on screen. Display the on-screen status indicator during the initial process of loading as well as during any later process

of loading a page, but do so only if the process takes more then 2 sec.” [Kunert 2009, p. 131]

→ Is almost identically with LOADING INDICATOR.

- SWITCHING BETWEEN CONTENT ITEMS: „Switching different content items usually contains two actions: Going one level up [...] then choosing the next item. The user task can be speeded up if all contents on one level are linked with each other.” [Kunert 2009, p. 142-143]

→ Is refined by SEQUENTIAL MEDIA.

- SYNCHRONISED CONTENT: „Give the user control over the presentation of temporally available contents and functions. Do not push the latter on the screen automatically. [...]” [Kunert 2009, p. 144-145]

→ Is abstracted by ANNOTATED TIMELINE and refined by OVERLAYS, CLOSED CAPTIONS, and ASSESSMENT.

- VOTING AND MULTIPLE CHOICE QUESTION: „For multiple choice questions provide 2-4 answer options. For opinion-polls provide 2-7 answer options.” [Kunert 2009, p. 148-149]

→ Is refined by ASSESSMENT and POLLS.

- ACCESSIBILITY: Adjustable font size; subtitles in different languages; audio feedback; avoid transparent background; [Kunert 2009, p. 168-169]

→ Is refined by CLOSED CAPTIONS and TRANSCRIPT.

## Patterns for Computer-Mediated Interaction

Source: Schümmer und Lukosch [2007]

Patterns: 72

- THREADED DISCUSSION: „Track and visualize the relations between the different messages.” [Schümmer und Lukosch 2007, p. 181-286]

→ Is abstracted by COMMENTS.

- ACTIVITY LOG: „Store information about users’ activities in a log to provide a history of their activities and the artifacts’ evolution.” [Schümmer und Lukosch 2007, p. 371-376]

→ Is identically with VIEWING HISTORY. Is refined by FOLLOW REVISIONS and USER TRACES.

## Patterns in Interaction Design

Source: [van Welie 2008; van Welie und Veer 2003] , <http://www.welie.com/patterns/> (last accessed February 5th 2015, pager disappeared a week after)

Patterns: 131

- COMMENT BOX: „Users want to comment on an article or product [...] Add a small form below the article and after the other comments that allows users to submit a comment.” <http://www.welie.com/patterns/showPattern.php?patternID=comment-box>

→ Is refined by COMMENTS.

- RATING: „The user wants rate an object she knowledge about [...] Present a rating next to the product and the option to rate it”

- <http://www.welie.com/patterns/showPattern.php?patternID=rating>  
→ Is identically with USER RATINGS.
- POLL: „Users want to state their opinion about a certain statement that is relevant to the site’s content [ . . . ] List the statements as exclusive options and present the results directly after voting.“  
<http://www.welie.com/patterns/showPattern.php?patternID=poll>  
→ Is refined by POLLS.
- SITE MAP: „The users need to find a specific page [ . . . ] Show a map of the site.“  
<http://www.welie.com/patterns/showPattern.php?patternID=site-map>  
→ Is abstracted by STORY GRAPH.
- DETAILS ON DEMAND: No descriptions available, but an example screenshot from <http://www.ns.nl/>.  
<http://www.welie.com/patterns/showPattern.php?patternID=details-on-demand>  
→ Is refined by DETAIL ON DEMAND.
- SLIDESHOW: „The user wants to view a series of images/photos [ . . . ] Show each image for some seconds and provide controls to manually navigate back and forward, pause/resume and stop/return.“  
<http://www.welie.com/patterns/showPattern.php?patternID=slideshow>  
→ Is abstracted by VISUAL SUMMARY.
- THUMBNAIL: „You want to give the user an idea of how an image, movie or page looks like, but doesn’t want to display the full-scale image, movie or page because of download time and screen space. [ . . . ] Display a smaller version of the movie, image or page.“  
<http://www.welie.com/patterns/showPattern.php?patternID=thumbnail>  
→ Is abstracted by VIDEO MANAGER and refined by VISUAL SUMMARY.

### Multimedia-Reuse Pattern Language

Source: [Cybulski und Linden 1998]

Patterns: 5

- GLUE: „Put the artefacts together and apply special glue to join them into a composite object. [ . . . ] All joined artefacts become the components of the composite artefact but they can be easily decoupled.“ [Cybulski und Linden 1998, p. 5]  
→ Is refined by MEDIA FRAGMENTS.
- DEFINE AND RUN PRESENTATION: „Plan the presentation of multimedia artefacts by organising them into a partial order using a special glue. Define a delivery channel that will carry the multimedia message as a series of artefacts. The choice of presented artefacts can be made either during the planning or delivery stage. The delivery can happen via a single channel or multiple channels.“ [Cybulski und Linden 1998, p. 9]  
→ Is refined by SIMULTANEOUS MEDIA, SYNCHRONIZED MAP, and SEQUENTIAL MEDIA.
- SYNCHRONISE CHANNELS: „For non-continuous artefacts define delay artefacts. Delays define the sequence of temporal positions within the composites of continuous and non-continuous artefacts. The solution supports multiway indexing.“ [Cybulski und Linden

1998, p. 10]

→ Is refined by SIMULTANEOUS MEDIA, SYNCHRONIZED MAP, and ANNOTATIONS.

- COMPONENTS LAYOUT: „Compose the artefacts into a layout, arrange them with the use of arrangement glue to associate each component with its position within the layout.“ [Cybulski und Linden 1998, p. 12]

→ Is refined by FULL SCREEN and OBJECT TRACKING.

## Yahoo Design Pattern Library

Source: Yahoo! [2012], <http://developer.yahoo.com/ypatterns/>

Patterns: 59

- DISPLAY COLLECTION: „Enabling people to display the objects they’ve collected contributes to the game-like dynamics of collecting, advertises to other users the types of things that may be collected and provides multiple cross-links from the person’s collection page to the details pages of the collected objects.“ [Yahoo! 2012]  
<https://developer.yahoo.com/ypatterns/social/objects/collecting/displaying.html>  
→ Is refined by PLAYLIST.
- FAVORITES: „Allow users to create a list of favorite objects on your site by choosing a "Favorite this" link closely associated to each object. Collect favorited items together on their own page for easy access, and associate the collected favorited items with the user’s profile or identity.“ [Yahoo! 2012]  
<https://developer.yahoo.com/ypatterns/social/objects/collecting/favorites.html>  
→ Is refined by PLAYLIST.
- RATING AN OBJECT: „Label a row of clickable items (often stars) that light up on rollover with invitational text above to invite the user to rate the object (e.g., "Rate It!"). The initial state should be empty. As the mouse cursor moves over the icons, display a text description of the rating at each point (e.g., "Excellent"). The saved rating should be indicated with a change in final color of the items and a text indication that the rating is saved.“ [Yahoo! 2012]  
<https://developer.yahoo.com/ypatterns/social/objects/feedback/rating.html>  
→ Is identically with USER RATINGS.
- TAG AN OBJECT: „Allow users to add their own tags (keywords) to an object. Be clear about how to separate (delimit) distinct tags. Consider allowing people to tag objects they did not create or don’t own, with the permission of the creator or owner. Don’t be afraid to mix a controlled vocabulary with user-generated tags.“ [Yahoo! 2012]  
<https://developer.yahoo.com/ypatterns/social/objects/collecting/tag/object.html>  
→ Is refined by TEMPORAL TAGS.

## Quince UX Paterns

Source: [Infragistics 2013], <http://quince.infragistics.com/html/Patterns>: 92

- ANNOTATED SCROLLBAR: „You have something, such as a lot of items in a list or a long document, in a scrollable area, and it is hard for people to use the scrollbar for navigation due to its length or size. [...] Provide an indicator of significant sections in or near the scrollbar.“ [Infragistics 2013]  
<http://quince.infragistics.com/html/PatternView.aspx?name=Annotated+Scrollbar>  
→ Is refined by ANNOTATED TIMELINE.
- EXTRAS ON DEMAND: „People need flexibility and power but don’t always need or want it in their faces. [...] Show the most common or important stuff first, and let people get to extra stuff easily.“ [Infragistics 2013]  
<http://quince.infragistics.com/html/PatternView.aspx?name=Extras+on+Demand>  
→ Is refined by DETAIL ON DEMAND.
- EDIT IN PLACE: „People need to edit some value as easily as possible. [...] Whenever possible, let people edit values in the same place that they are displayed.“ [Infragistics 2013]  
<http://quince.infragistics.com/html/PatternView.aspx?name=Edit-in-Place>  
→ Is abstracted by DIRECT AUTHORIZING
- INPUT PROMPT: „Sometimes people don’t know what to put into fields on forms. [...] Supply an explanation or example of what a person should do with an input inside of the input itself.“ [Infragistics 2013]  
<http://quince.infragistics.com/html/PatternView.aspx?name=Input+Prompt>  
→ Is abstracted by DISCUSSION PROMPTS
- JOURNAL NAVIGATION: „People tend to think linearly and often like to be able to navigate that way. [...] Give people a mechanism that they can use to go back and forth in their navigation history.“ [Infragistics 2013]  
<http://quince.infragistics.com/html/PatternView.aspx?name=Journal+Navigation>  
→ Is abstracted by JOURNAL NAVIGATION
- LOCAL ZOOMING: „People want details when they want them without losing context. [...] Show an overview of your information with the ability to select specific points to see more details without losing the context.“ [Infragistics 2013]  
  
→ Is abstracted by DETAIL ON DEMAND and refined by *Zoom*.

## Patterns for Time-based Hypermedia Artifacts

Source: [Lopes und Carriico 2007]  
Patterns: 21

- TIME COMPOSITION: „[...] provide different time composition constructs, either through absolute timestamps (e.g., play X at 01m:02s) or higher level relations (e.g., play Y after X), therefore supporting both quantitative and qualitative time references [...]“ [Lopes

- und Carriico 2007]
- Is refined by SIMULTANEOUS MEDIA.
  - TIMELINE: „Provide timeline capabilities in the time-based hypermedia system, related to the presentation overall duration. When“ [Lopes und Carriico 2007, p. 9]
    - Is abstracted by BASIC CONTROLS and refined by ANNOTATED TIMELINE.
  - CONTENT DECOUPLING: „The system developer should provide a data structure that allows the decoupling between all content parts.“ [Lopes und Carriico 2007, p. 100-11]
    - Is related to REMIX.
  - MULTIPLE CONTENT SOURCES: „Provide different media sources regarding the same content structure, by providing an additional layer of indirection between the content structure and the actual related media.“ [Lopes und Carriico 2007, p. 12]
    - Is refined by TRANSCRIPT and CLOSED CAPTION.
  - INTEGRATED NAVIGATIONAL CONTEXTS: „Provide different navigation contexts for the content. When the time flow changes (either by time shifts specified in the application or triggered by the user through some navigation mechanism), all navigation structures should reflect this change accordi[n]gly, in a synchronous way.“ [Lopes und Carriico 2007, p. 13]
    - Is related to TABLE OF CONTENT, TEMPORAL TAGS, and ANNOTATIONS in general.
  - TIMED ACTIVE REFERENCE: „As time-based content presentation is inherently dynamic, navigation structures must be updated constantly by presenting the current navigation status within each navigation structure, whether if triggered by user events or just through the application's time flow. If“ [Lopes und Carriico 2007, p. 14]
    - Is almost identically with VIEWING HISTORY and refined by FOLLOW REVISIONS.
  - LINK OPPORTUNITIES: „The volatile nature of time limited links easily creates stress to the end user, as link opportunities may vanish before the user's decision on following the link.“ [Lopes und Carriico 2007, p. 15]
    - Is related to HYPERLINK and OVERLAYS.
  - TIMEWARP LINKING: „How to update the navigation state of the link's destination content, if there is no purpose to get back to the link's source content? [...] Perform a shift in the presentation's current time, stopping the playback in the source anchor context and starting the playback in the destination anchor [...].“ [Lopes und Carriico 2007, p. 16]
    - Is related to HYPERLINKS.
  - MONTAGE TIME LINKING: „You have created a link between a content item and some complementary content, in the time-based hypermedia presentation. [...] Allow an expansion on the presentation towards two distinct contents [...]. Both contents must be presented to the user at the same time.“ [Lopes und Carriico 2007, p. 17]
    - Is related to SIMULTANEOUS MEDIA.
  - TIMEBREAK LINKING: „How should navigation mechanisms be defined, that they will minimize the cognitive effort of presenting two contents, yet allowing the user to get back to the previous presentation's state? [...] Pause the playback of the source content, and begin the playback of the target content. Afterwards, resume the source content [...].“ [Lopes und Carriico 2007, p. 18]
    - Is related to DETAIL ON DEMAND.



- NEIGHBOURHOOD LINKING: „How to allow time-based navigation to behave like traditional forward/back navigation mechanisms? [...] Provide a navigation structure related to the beginning time instance of each content part that will behave like forward/back mechanisms, through associated interaction mechanisms (such as back/forward buttons).“ [Lopes und Carriico 2007, p. 19]
  - Is almost identical with SEQUENTIAL MEDIA.
- STRICT SYNCHRONIZATION PRESENTATION: „You are presenting two synchronized contents, strictly correlated, possibly pertaining to two expressions of the same content (e.g., a video and its subtitles, an animation and a complementary text, etc.). [...] Provide lesser strict synchronization guides, that will help the user to know (with a good certainty level) which part of the content is currently being played [...]“ [Lopes und Carriico 2007, p. 21]
  - Is related to TRANSCRIPT, TABLE OF CONTENT, and SYNCHRONIZED MAP.
- LINK OPPORTUNITY MANIFESTATION: „You want to present to the user a link opportunity in the time-based hypermedia presentation. [...] How to make perceivable to the user (in time-based mediums) that a link is nearby, currently active, or going to disappear? [...] On the visual medium, markers can be placed nearby the timeline, identifying a link opportunity's starting and ending points [...]“ [Lopes und Carriico 2007, p. 22]
  - Is abstracted by ANNOTATED TIMELINE, but related to HYPERLINKS and OVERLAYS.
- TIMED INFORMATION ON DEMAND: „You want to present to the user a complementary content, regarding the current presentation content. [...] How to minimize the cognitive problems that arise when complex contents are presented at the same time, even if these are complementary? [...] Present to the user a simplified version of the complementary content, as well as the opportunity to reach its full content with a link opportunity. Make sure that the simplified version's interference is smaller than its full content counterpart (e.g., text caption vs. video).“ [Lopes und Carriico 2007, p. 23]
  - Is almost identically with DETAIL ON DEMAND.
- MULTIPLE CHANNEL PRESENTATION: „How to avoid possible cognitive overloads that may appear when playing two or more contents? [...] Present multiple content sources to the user as long as the cognitive effort required to capture each content does not create cognitive pressure. Each media type requires different efforts, thus being reflected in media composition through different channels [...]“ [Lopes und Carriico 2007, p. 25]
  - Is related to SIMULTANEOUS MEDIA and any kind of OVERLAYS.
- CONTEXT ON USER EVENT: „When jumping into a random location in the timeline, the user may be confused by the content's presentation, as it may appear out of context (e.g., middle of a text paragraph, end of a spoken phrase on an audio track, etc.). [...] Provide the user enough content to enable contextualization (e.g., a whole text phrase or paragraph, an overview of a full picture, rewind a time-based content a few seconds, etc.).“ [Lopes und Carriico 2007, p. 27]
  - Is related to BASIC CONTROLS and TEMPORAL TAGS but abstracts SKIP BACK.

**Zusammenfassung**

In Tab. A.14 sind die Ergebnisse der Analyse verwandter Muster zusammenfassend dargestellt.

**Tabelle A.14:** Art der Beziehung zwischen den identifizierten Mustern und den Mustern in anderen Musterkollektionen. (A = Abstraktion, C = Komplementäres Muster, I = fast identisches Muster, R = Verfeinerung des identifizierten Musters, L = Verbund)

Design Pattern	[Tidwell 2005, 2011]	[Kunert 2009]	[Schümmer/Lukosch 2007]	[van Welie und Veer 2003]	[Cybulski und Linden 1998]	[Yahoo! 2012]	[Infragistics 2013]	[Lopes und Carrico 2007]	Anzahl an Beziehungen
ADD VIDEO	-	-	-	-	-	-	-	-	0
ANNOTATED TIMELINE	R	A	-	-	-	-	R	RA	5
ANNOTATIONS	-	-	-	-	R	-	-	L	2
APPROPRIATE DELIVERY	-	-	-	-	-	-	-	-	0
ASSESSMENT	-	R	-	-	-	-	-	-	1
BASIC CONTROLS	-	-	-	-	-	-	-	AL	2
BRANCHING VIDEOS	-	-	-	-	-	-	-	-	0
CLASSIFIED MARKS	-	-	-	-	-	-	-	-	0
CLOSED CAPTIONS	-	R	-	-	-	-	-	R	2
COMMENTS	-	-	A	R	-	-	-	-	2
DETAIL ON DEMAND	R	-	-	R	-	-	AR	LI	6
DIRECT AUTHORING	A	-	-	-	-	-	R	-	2
DISCUSSION PROMPT	R	-	-	-	-	-	R	-	2
FULL SCREEN	-	-	-	-	R	-	-	-	1
FOLLOW REVISIONS	R	-	R	-	-	-	-	R	3
HYPERLINKS	-	-	-	-	-	-	-	L	1
JOURNALED NAVIGATION	-	-	-	-	-	-	A	-	1
LOADING INDICATOR	I	I	-	-	-	-	-	-	2
LOOP	-	-	-	-	-	-	-	-	0
MEDIA FRAGMENTS	-	-	-	-	R	-	-	-	1
MULTI-ANGLE VIDEO	R	I	-	-	-	-	-	-	2
MULTI-TIMELINE EDITING	-	-	-	-	-	-	-	-	0
OBJECT TRACKING	-	-	-	-	R	-	-	-	1
OVERLAYS	R	R	-	-	-	-	-	L	3
PLAYBACK SPEED	-	-	-	-	-	-	-	-	0
PLAYBACK DIRECTION	-	-	-	-	-	-	-	-	0
PLAYLIST	-	-	-	-	-	R	-	-	1
POLLS	-	R	-	R	-	-	-	-	2
RELATED VIDEOS	-	-	-	-	-	-	-	-	0
REMIX	-	-	-	-	-	-	-	L	1
SEARCH	-	-	-	-	-	-	-	-	0
SEQUENTIAL MEDIA	CR	R	-	-	R	-	-	I	5
SHORTCUT COMMANDS	R	R	-	-	-	-	-	-	2
SIMULTANEOUS MEDIA	-	R	-	-	R	-	-	RL	4
SKIP BACK	-	-	-	-	-	-	-	L	0
STORY GRAPH	C	-	-	A	-	-	-	-	2
SYNCHRONIZED MAPS	-	-	-	-	R	-	-	L	2
TABLE OF CONTENT	-	R	-	-	-	-	-	L	2
TEMPORAL BOOKMARKS	-	-	-	-	-	-	-	-	0
TEMPORAL TAGS	-	R	-	-	-	R	-	L	3
TRANSCRIPT	-	-	-	-	-	-	-	RL	2
USER NOTES	-	-	-	-	-	-	-	-	0
USER RATINGS	-	-	-	R	-	R	-	-	2
USER TRACES	-	-	R	-	-	-	-	-	1
VIDEO MANAGER	R	A	-	A	-	-	-	-	3
VIDEO MANIPULATION	-	-	-	-	-	-	-	-	0
VIEWING HISTORY	-	-	I	-	-	-	-	I	2
VISUAL HIGHLIGHTING	-	R	-	-	-	-	-	-	1
VISUAL SUMMARY	R	-	-	AR	-	-	-	-	3
ZOOM	R	-	-	-	-	-	R	-	2
Anzahl an Beziehungen	15	14	4	8	7	3	7	22	80

## Ergebnisse der Pattern-Evaluation

Patterns	Richtigkeit				Vollständigkeit			
	TN1	TN2	TN3	AVG	TN1	TN2	TN3	AVG
Add Video	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Annotated Timeline	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Annotations	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Appropriate Delivery	10,00	10,00	10,00	10,00	10,00	10,00	5,00	8,33
Assessment	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Basic Controls	10,00	10,00	10,00	10,00	5,00	10,00	10,00	8,33
Branching Video	0,00	10,00	10,00	6,67	0,00	8,00	10,00	6,00
Classified Marks	10,00	0,00	5,00	5,00	10,00	0,00	5,00	5,00
Closed Captions	10,00	5,00	10,00	8,33	5,00	10,00	10,00	8,33
Comments	10,00	10,00	10,00	10,00	10,00	8,00	10,00	9,33
Detail on Demand	10,00	10,00	5,00	8,33	10,00	10,00	5,00	8,33
Direct Authoring	10,00	10,00	10,00	10,00	10,00	5,00	5,00	6,67
Follow Revision	10,00	10,00	10,00	10,00	10,00	10,00	8,00	9,33
Full Screen	10,00	10,00	5,00	8,33	8,00	5,00	5,00	6,00
Hyperlinks	0,00	0,00	10,00	3,33	0,00	0,00	5,00	1,67
Journaled Navigation	5,00	10,00	10,00	8,33	5,00	10,00	5,00	6,67
Loading Indicator	10,00	10,00	10,00	10,00	5,00	10,00	5,00	6,67
Loop	10,00	5,00	10,00	8,33	10,00	10,00	10,00	10,00
Media Fragments	10,00	10,00	10,00	10,00	10,00	8,00	10,00	9,33
Multi-Timeline Editing	10,00	10,00	0,00	6,67	10,00	10,00	0,00	6,67
Object Tracking	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Overlays	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Playback Speed	10,00	10,00	10,00	10,00	10,00	5,00	10,00	8,33
Playlist	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Related Videos	10,00	10,00	10,00	10,00	10,00	5,00	5,00	6,67
Remix	0,00	0,00	10,00	3,33	0,00	0,00	10,00	3,33
Search	5,00	10,00	10,00	8,33	5,00	10,00	10,00	8,33
Sequential Media	5,00	0,00	10,00	5,00	5,00	0,00	10,00	5,00
Shortcut Commands	10,00	5,00	10,00	8,33	10,00	10,00	10,00	10,00
Simultaneous Media	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Skip Back	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Synchronized Map	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Temporal Bookmarks	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Temporal Tags	10,00	10,00	5,00	8,33	10,00	10,00	5,00	8,33
Table of Content	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Transcript	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
User Notes	10,00	10,00	0,00	6,67	10,00	10,00	0,00	6,67
User Ratings	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
User Traces	10,00	5,00	10,00	8,33	10,00	5,00	10,00	8,33
Video Manager	10,00	10,00	10,00	10,00	10,00	10,00	8,00	9,33
Video Manipulation	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Viewing History	0,00	10,00	10,00	6,67	0,00	10,00	10,00	6,67
Visual Highlighting	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
Visual Summary	0,00	10,00	10,00	6,67	0,00	10,00	10,00	6,67
Zoom	0,00	10,00	10,00	6,67	0,00	10,00	10,00	6,67
Summe	355,00	370,00	390,00	371,67	338,00	359,00	356,00	351,00
Arithmetisches Mittel	8,07	8,41	8,86	8,45	7,68	8,16	8,09	7,98
Standardabweichung	3,77	3,37	2,83	2,28	3,79	3,37	3,04	2,36

**Abbildung A.4:** Evaluationsergebnisse der von drei Teilnehmern zu jedem Pattern eingereichten Skizzen für eine mögliche Umsetzung der im Pattern beschriebenen Lösung. Alle Patterns, die bezüglich der Bewertungsdimensionen Richtigkeit und Vollständigkeit weniger als 8 Punkte erhielten, wurden noch einmal grundlegend überarbeitet.

## Kategorisierung der Patterns nach dem Universal Model of User Interface

**Tabelle A.15:** Den Patterns zugeordnete Layer des *Universal Model of User Interface* [Baxley 2003].

Pattern	Zugeordnetes Layer
Add Video	Editing and Manipulation
Annotated Timeline	Organizational Model
Annotations	Editing and Manipulation
Appropriate Delivery	Layout
Assessment	Editing and Manipulation
Basic Controls	Viewing and Navigation
Branching Videos	Viewing and Navigation
Classified Marks	Organizational Model
Closed Captions	User Assistance
Comments	Editing and Manipulation
Detail on Demand	Viewing and Navigation
Direct Authoring	Editing and Manipulation
Follow Revisions	Task Flow
Full Screen	Layout
Hyperlinks	Viewing and Navigation
Journaled Navigation	Viewing and Navigation
Loading Indicator	User Assistance
Loop	Viewing and Navigation
Media Fragments	Editing and Manipulation
Multi-Timeline Editing	Editing and Manipulation
Object Tracking	Viewing and Navigation
Overlays	Viewing and Navigation
Playback Speed	Viewing and Navigation
Playlist	Organizational Model
Related Videos	Organizational Model
Remix	Editing and Manipulation
Search	Viewing and Navigation
Sequential Media	Task Flow
Shortcut Commands	Viewing and Navigation
Simultaneous Media	Layout
Skip Back	Viewing and Navigation
Synchronized Map	Viewing and Navigation
Table of Content	Organizational Model
Temporal Bookmarks	Organizational Model
Temporal Tags	Organizational Model
Transcript	User Assistance
User Notes	Editing and Manipulation

(Fortsetzung auf der nächsten Seite)

(Tabelle A.15 – Fortsetzung von der vorherigen Seite)

User Ratings	Organizational Model
User Traces	Organizational Model
Video Manager	Task Flow
Video Manipulation	Editing and Manipulation
Viewing History	Viewing and Navigation
Visual Highlighting	Viewing and Navigation
Visual Summary	Viewing and Navigation
Zoom	Viewing and Navigation

## Entscheidungsmodell zur Auswahl der Design Patterns

**Tabelle A.16:** Binäres Entscheidungsmodell zur Suche und Auswahl von Design Patterns innerhalb der in Abschnitt 4.3.3 auf S. 145 vorgestellten Wissensbasis. In der Matrix sind den Patterns die 12 Fragen (F1–F12) gegenübergestellt. Die Disjunktion der Wahrheitswerte der einzelnen Fragen ergeben die aussagenlogische Formel in der letzten Spalte.

Design Pattern	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	¬F12	Aussagenlog. Notation
ADD VIDEO	0	0	0	0	0	0	0	1	0	0	0	0	$F8$
ANNOTATED TIMELINE	0	0	0	0	1	0	0	0	0	0	0	1	$F5 \vee F12$
ANNOTATIONS	0	0	0	0	0	0	0	1	0	1	0	0	$F8 \vee F10$
ASSESSMENT	0	0	0	0	0	0	0	0	1	1	0	0	$F9 \vee F10$
BRANCHING VIDEOS	0	1	0	0	1	0	0	0	0	0	0	0	$F5$
CLASSIFIED MARKS	0	0	0	0	0	0	0	0	0	0	0	1	$F12$
CLOSED CAPTIONS	0	0	0	1	0	0	0	0	0	0	0	0	$F4$
COMMENTS	0	0	0	0	0	0	0	1	0	1	1	0	$F8 \vee F10 \vee F11$
DETAIL ON DEMAND	0	1	0	0	1	0	0	0	0	0	0	0	$F5$
DIRECT AUTHORIZING	0	0	0	0	0	0	0	1	0	0	0	0	$F8$
FOLLOW REVISIONS	0	0	0	0	0	0	0	1	0	0	0	0	$F8$
FULL SCREEN	0	0	1	0	0	0	0	0	0	0	0	0	$F3$
HYPERLINKS	0	1	0	0	1	0	0	0	0	0	0	0	$F5$
JOURNALED NAVIGATION	0	1	0	0	1	0	0	0	0	0	0	0	$F5$
LOOP	0	0	0	0	0	0	1	0	0	0	0	0	$F7$
MEDIA FRAGMENTS	0	0	0	0	0	0	0	1	0	0	0	0	$F8$
MULTI-TIMELINE EDITING	0	0	0	0	0	0	0	1	0	0	0	1	$F8 \vee F12$
OBJECT TRACKING	0	0	0	0	0	0	1	1	0	0	0	0	$F7 \vee F8$
OVERLAYS	0	0	0	0	0	0	0	0	0	0	0	1	$F12$
PLAYBACK SPEED	0	0	0	0	0	0	1	0	0	0	0	0	$F7$
PLAYLIST	0	1	0	0	0	0	0	0	1	0	0	0	$F2 \vee F9$
RELATED VIDEOS	0	1	0	0	0	0	0	0	1	0	0	0	$F2 \vee F9$
REMIX	0	0	0	0	0	0	0	1	0	0	0	1	$F8 \vee F11$
SEARCH	0	0	0	0	0	0	0	0	1	0	0	0	$F9$
SEQUENTIAL MEDIA	1	0	0	0	0	0	0	0	0	0	0	0	$F1$
SIMULTANEOUS MEDIA	0	0	1	0	0	0	0	0	0	0	0	1	$F3 \vee F11$
SKIP BACK	0	0	0	0	0	0	1	0	0	0	0	0	$F7$
SYNCHRONIZED MAP	0	0	1	0	0	0	0	0	0	0	0	1	$F3 \vee F12$
TABLE OF CONTENT	0	0	0	0	0	1	0	0	1	0	0	0	$F6$
TEMPORAL BOOKMARKS	0	0	0	0	0	0	0	0	1	0	0	0	$F9$
TEMPORAL TAGS	0	0	0	0	1	1	0	0	0	0	0	0	$F2$
TRANSCRIPT	0	0	0	1	0	0	0	0	0	0	0	0	$F4$
USER NOTES	0	0	0	0	0	0	0	1	1	0	0	0	$F8 \vee F9$
USER RATINGS	0	1	0	0	0	0	0	0	0	1	1	0	$F2 \vee F10 \vee F11$
USER TRACES	0	0	0	0	0	0	0	0	1	0	1	1	$F9 \vee F11 \vee F12$
VIDEO MANAGER	0	1	0	0	0	0	0	0	0	0	0	0	$F2$
VIDEO MANIPULATION	0	0	0	0	0	0	0	1	0	0	0	1	$F8 \vee F12$
VIEWING HISTORY	0	1	0	0	0	0	0	0	1	0	0	0	$F2 \vee F9$
VISUAL HIGHLIGHTING	0	0	0	0	0	0	1	0	0	0	0	0	$F7$
VISUAL SUMMARY	0	1	0	0	0	1	0	0	0	0	0	0	$F2 \vee F6$
ZOOM	0	0	0	0	0	0	1	0	0	0	0	0	$F7$



## Funktionale Analyse der Videolernumgebung eKomMed mit Hilfe der Design Patterns

*eKomMed* ist ein Lehr-Lern-Projekt, welches im Rahmen des Projekts *Lehrpraxis im Transfer* durch das Hochschuldidaktische Zentrum Sachsen durch Mittel aus dem *Qualitätspakt Lehre* (BMBF) gefördert wurde.

**Kontext der Videolernumgebung eKomMed** Das Hauptziel des Projekts *eKomMed*<sup>21</sup> bestand im „Aufbau und Einsatz eines Online-Videoportals zur Unterstützung des Erwerbs kommunikativer Kompetenzen im Medizinstudium“ [Rockenbach und Cämmerer 2014] an der Universität Leipzig sowie an der Technischen Universität Dresden. Die Videolernumgebung<sup>22</sup> besteht aus acht Gesprächsvideos zwischen Arzt und Patienten, die jeweils um ein Diskussionsforum angereichert wurden. Die didaktische Zielsetzung hinter der Lernumgebung ist die Schaffung eines Zusatzangebots zur Präsenzlehre in der Medizin, welches Studierende im Selbststudium nutzen können. Die Lernumgebung soll Studierende dazu anregen, ihr eigenes Verhalten in Gesprächen mit Patienten zu reflektieren. Im Detail „bietet [die Lernumgebung] die Möglichkeit zur Verbesserung der eigenen Gesprächsführung durch gute Beispiele bzw. durch Reflexion und Diskussion ‘schlechter’ Gesprächsbeispiele“ [Rockenbach und Cämmerer 2014]. Ein Teilziel des Projekts war es herauszufinden, wie Studierende die produzierten Videos bewerten und, zu welchem persönlichen Lernerfolg sie ihnen verhelfen. Darüber hinaus dient die Anwendung zur Vorbereitung auf eine Prüfung (Universität Leipzig), in der Studierende in einem Rollenspiel mit fiktiven Patienten ihre Fähigkeiten in der Gesprächsführung unter Beweis stellen müssen. Die Zielgruppe an beiden Hochschulen ist in Bezug auf Sprachkenntnisse, Fachsemester und Vorkenntnisse sehr heterogen.

**Evaluation** Die Evaluation der online verfügbaren Lernumgebung wurde am 23. Oktober 2013 durch den Autor gemäß der auf S. 149 definierten Schritte vorgenommen.

- **Schritt 1 – Kontext ermitteln:** Ausgangspunkt bilden die Kontextvariablen, die eine Videolernumgebung charakterisieren. Diese werden mit Hilfe des Entscheidungsmodells in Form des Wizzards (siehe 4.3.3) ermittelt. Aus den 11 Fragen im Wizzard resultieren die 25 in Tab. A.17 gelisteten Design Patterns.
- **Schritt 2 – Abgleich mit Lernumgebung:** Auf acht Unterseiten der Lernumgebung waren Videos integriert. Hinsichtlich der Seitenstruktur waren diese Seiten fast identisch. Es konnten abgesehen von den Grundlegenden Patterns (BASIC CONTROLS, KEYBOARD COMMANDS, LOADING INDICATOR) keines der in Schritt 1 ermittelten Patterns identifiziert werden. Dennoch finden sich einige Lösungsansätze für z.B. COMMENTS oder ASSESSMENT.
- **Schritt 3 – Verbesserungsvorschläge:**
  - Abgesehen von den Grundfunktionen BASIC CONTROLS, KEYBOARD COMMANDS, LOADING INDICATOR wurde kein einziges im Wizzard identifiziertes Pattern in *eKomMed* vollständig implementiert.

<sup>21</sup>Der vollständige Projektname lautet: „Einführung eines standortübergreifenden e-learning tools zum Erlernen kommunikativer Kompetenzen in der Medizin“

<sup>22</sup>Siehe <http://www.ekommed.de/> (abgerufen am 30.08.2014).

- Es ließen sich keine Gestaltungslösungen identifizieren, die über die in der Pattern Language festgehaltenen Lösungen hinausgehen.
- Mangels identifizierter Muster, ließen sich auch keine alternativen Realisierungen ermitteln.

In Tab. A.17 sind die Evaluationsergebnisse von *eKomMed* dargestellt. Je ermittelten Pattern wird kommentiert, in wie weit die darin enthaltene Gestaltungslösung in der Videolernumgebung realisiert wurde und, ob diese in Anbetracht des Videomaterials, technischer Limitationen und der didaktischen Zielstellung überhaupt erforderlich ist. Die vorgenommene Priorisierung von 1 (hohe Priorität) bis 3 (niedrige Priorität) drückt aus, welche Interaction Design Lösungen gemäß der Patterns bevorzugt umgesetzt werden sollten. Im vorliegenden Fall betrifft dies die Pattern COMMENTS, PLAYBACK SPEED, USER NOTES, USER RATINGS und *Video Manager*. Insgesamt wurde die Lernumgebung hinsichtlich des Interaktionsdesigns zum Zeitpunkt der Untersuchung als stark verbesserungswürdig eingeschätzt. Die Anregungen und Verbesserungsvorschläge konnten von den beteiligten Projektmitarbeitern, angesichts begrenzter Personalkapazitäten bei der Entwicklung, zumindest teilweise berücksichtigt werden.

**Tabelle A.17:** Evaluationsergebnisse und Bewertung der Videolernumgebung *eKomMed*.

Patterns je Wizzard-Frage	Evaluationsergebnis und Kommentar	Priorität
<b>F1 – Mehrere Medien</b>		
SEQUENTIAL MEDIA	Implementiert. Die acht Videos sind in einer Reihe arrangiert und als solche im Menü sichtbar, wobei man nicht einfach vorwärts und rückwärts schalten kann.	–
<b>F2 – Mehr als ein Video</b>		
VIDEO MANAGER	Nicht implementiert. Neben dem Menü der acht Videos fehlt eine Übersichtsseite, auf der Themen und Schwerpunkte der einzelnen Videos nebst einem repräsentativen Bild zu sehen sind. Acht verschiedene Elemente können Nutzer in der Regel nicht memorieren. Eine solche Übersichtsseite wäre zudem ein passenderer Einstieg, als die Projektbeschreibung.	1
RELATED VIDEOS	Nicht implementiert. Um die Anwender zur Bearbeitung weiterer Videos zu motivieren, wäre es wünschenswert thematisch ähnliche Arzt-Patienten-Gespräche zu empfehlen oder zumindest eine alternative Sequenzierung als im Menü aufzuzeigen.	2

(Fortsetzung auf der nächsten Seite)

(Tabelle A.17 – Fortsetzung von der vorherigen Seite)

PLAYLIST	Nicht implementiert, ist jedoch angesichts der noch überschaubaren Anzahl von acht Videos nicht unbedingt notwendig.	3
USER RATINGS	Nicht implementiert, entspricht jedoch einer der didaktischen Zielsetzungen (und auch Forschungsfrage). Studierende sollten mitteilen können, welche Videos ihrer Meinung nach die Wissensaneignung befördern.	1
VIEWING HISTORY	Nicht implementiert. Im Kontext des selbstregulierten Lernens erscheint es sinnvoll, Studierenden mitzuteilen, in wie weit sie einzelne Videos schon betrachtet und bearbeitet haben.	2
VISUAL SUMMARY	Mangels VIDEO MANAGER nicht implementiert. Ein Vorschau auf Szenen könnte helfen, innerhalb eines Videos zu navigieren.	2

**F3 – Synchrone Mediendarstellung**

SIMULTANEOUS MEDIA	Nicht implementiert. Des öfteren sind mehrere Kameraperspektiven in einem Videobild vereint. Diese Auswahl der Produzenten könnte auch durch separate Medienströme je Kameraeinstellung realisiert werden, die der Anwender nach belieben zu- und abschalten kann.	3
SYNCHRONIZED MAPS	Nicht implementiert, da die Lernressourcen keine Karten enthalten.	–
FULL SCREEN	Nicht implementiert, da es auch keine simultan dargebotenen Medien gibt.	1

**F4 – Barrierefreiheit**

CLOSED CAPTIONS	Nicht implementiert. Untertitel könnten Nicht-Muttersprachlern das Verständnis erleichtern und zudem die Durchsuchbarkeit der Inhalte verbessern. Im Hinblick auf die Prüfung in einem Rollenspiel, trägt die Abstinenz dieses Patterns zur Authentizität der im Video dargestellten Situation bei.	3
-----------------	---	---

(Fortsetzung auf der nächsten Seite)

(Tabelle A.17 – Fortsetzung von der vorherigen Seite)

TRANSCRIPT	Nicht implementiert. Ein Transkript des gesprochenen Wortes könnte Nicht-Muttersprachlern das Verständnis erleichtern und zudem die Durchsuchbarkeit der Inhalte verbessern. Im Hinblick auf die Prüfung in einem Rollenspiel, trägt die Abstinenz dieses Patterns zur Authentizität der im Video dargestellten Situation bei.	3
------------	--	---

#### **F5 – Assoziationen zwischen Videos**

(nicht vorhanden)

#### **F6 – Videos länger als 10 Minuten**

(nicht vorhanden)

#### **F7 – Videos mit hoher Dynamik bzw. Detailgehalt**

PLAYBACK SPEED	Nicht implementiert. Angesichts des hohen Detailgehalts der Videos (insbesondere Mimik und Gestik der Darsteller) wäre eine verlangsamte Wiedergabe eine hilfreiche Unterstützung bei der Bearbeitung der Aufgabenstellung.	1
SKIP BACK	Nicht implementiert. Die Möglichkeit für einige Sekunden auf der Zeitachse des Videos zurück zu springen wäre hilfreich, um Aussagen im Dialog zwischen Arzt und Patient noch einmal zu wiederholen. Die gebotene Zeitleiste des Players ermöglicht keine präzise, d.h. sekundengenaue Navigation.	2
VISUAL HIGHLIGHTING	Nicht implementiert. Die Hervorhebung von bestimmten Bildbereichen wäre in Einzelfällen in Kombination mit einer zeitgenauen Kommentarfunktion sinnvoll.	3

(Fortsetzung auf der nächsten Seite)

(Tabelle A.17 – Fortsetzung von der vorherigen Seite)

ZOOM	Nicht implementiert. Zur Unterstützung der Analyse von Mimik und Gestik der Darsteller im Video würde sich eine Zoom-Funktion anbieten. Der Detailgehalt der einzelnen Videos ist teilweise sehr hoch. Durch die parallel Darstellung zweier Kameraperspektiven sind die Gesichter der Darsteller nicht in voller Größe sichtbar.	2
LOOP	Nicht implementiert, in Anbetracht des Videomaterials und der didaktischen Zielsetzung nicht erforderlich.	–
OBJECT TRACKING	Nicht implementiert und aufgrund der geringen Dynamik des Bildes bei Abstinenz von Overlays nicht erforderlich.	–

### **F8 – Inhaltliche Erweiterung der Videos**

(nicht vorhanden)

### **F9 – Selbstgesteuertes Lernen**

ASSESSMENT	Teilweise implementiert im Sinne einer zeitunabhängigen Überprüfungsaufgabe. Angesichts der Aufgabenstellung, den Arzt-Patienten-Dialog zu diskutieren, erscheint die Lösungsalternative zeitgenauer Fragen nicht unbedingt erforderlich.	3
PLAYLIST	(siehe oben)	
RELATED VIDEOS	(siehe oben)	
SEARCH	Nicht implementiert. Eine Suche nach Titel, Schlagwörtern oder innerhalb der Foren wäre wünschenswert, um eine effizienter Navigation zu ermöglichen. Gleiches gilt für die Dialoge der Darsteller.	3

(Fortsetzung auf der nächsten Seite)

(Tabelle A.17 – Fortsetzung von der vorherigen Seite)

TEMPORAL BOOKMARKS	Nicht implementiert. Um im Forum oder an anderer Stelle auf eine bestimmte Position in einem Video hinweisen und verweisen zu können, sind zeitgenaue Lesezeichen erforderlich. Mit Hilfe von Lesezeichen können Lernende ihren Lernprozess strukturieren und unnötige Zeit für die Suche nach bestimmten Szenen oder Gesprächsthemen zwischen Arzt und Patient vermeiden.	2
USER NOTES	Nicht implementiert. Persönliche Notizen in Abhängigkeit der Wiedergabezeit eines Videos helfen Lernenden die Wissensaneignung zu strukturieren oder zentrale Inhalte zusammenzufassen. Insbesondere zur Wiederholung der Aufgaben (z.B. zur Prüfungsvorbereitung) ist es sinnvoll auf Notizen zuzugreifen zu können, die unmittelbar neben dem Video hinterlegt sind und darüber hinaus eine individuelle zeitgenaue Navigation ermöglichen.	1
USER TRACES	Nicht implementiert. In Anbetracht des kollaborativen Nutzungsszenarios, erscheint es sinnvoll sowohl die individuell betrachteten Abschnitte des Videos entlang der Zeitleiste hervorzuheben, als auch jene Passagen, die andere Studierenden häufig rezipiert haben. Letzteres kann die Diskussion im Forum anregen.	2
VIEWING HISTORY	Nicht implementiert, jedoch ist es in Anbetracht der überschaubaren Anzahl an Videos nicht unbedingt erforderlich, die zuletzt betrachteten Videos anzuzeigen.	3
<b>F10 – Rückkanal</b>		
ANNOTATIONS	Nicht implementiert. Mit Ausnahme der nachfolgend erwähnten zeitgenauen Kommentare, erscheinen weitere Formen von Annotation in Anbetracht des Videomaterials und der didaktischen Zielsetzung nicht erforderlich.	1

(Fortsetzung auf der nächsten Seite)

(Tabelle A.17 – Fortsetzung von der vorherigen Seite)

COMMENTS	Teilweise implementiert. Die Kommentare zu vordefinierten Fragestellungen können unterhalb des Videos in einem Forum hinterlegt werden. Die Foreneinträge sind jedoch losgelöst vom Videoinhalt, auf den sie sich beziehen. Eine Verlinkung zwischen Text und Abspielposition ist ebenso wenig vorgesehen, wie Hinweise auf bestehende Foreneintrag während der Betrachtung des Videos.	1
USER RATINGS	(siehe oben)	
ASSESSMENT	(siehe oben)	

**F11 – Interaktion mit Peers**

COMMENTS	(siehe oben)	
SIMULTANEOUS MEDIA	(siehe oben)	
REMIX	Nicht implementiert, entspricht jedoch auch nicht der didaktischen Zielsetzung.	–
USER RATINGS	(siehe oben)	
USER TRACES	(siehe oben)	

**F12 – Verwendung mit Geräten mit kleinem Display**

(nicht vorhanden)



## PLML DTD und Mongoose Schema

Das Datenformat der Patterns innerhalb der Pattern Language orientiert sich an dem von Fincher [2003] vorgeschlagenen Format der *Pattern Language Markup Language* (PLML). In der Auflistung A.1 ist die dafür vorgesehenen *Document Type Definition* dargestellt. Die Wissensbasis liefert die Daten darüber hinaus als *JavaScript Object Notation* (JSON) aus. Das zugrundeliegende Schema der *MongoDB*-Datenbankschnittstelle *Mongoose* ist in der Auflistung A.18 zu sehen. Das Schema wurde lediglich um ein Feld für Kommentare seitens der Pattern-Autoren erweitert.

```

1 <!ELEMENT pattern (
2   name?,
3   alias*,
4   illustration?,
5   problem?,
6   context?,
7   forces?,
8   solution?,
9   synopsis?,
10  diagram?,
11  evidence?,
12  confidence?,
13  literature?,
14  implementation?,
15  related-patterns?,
16  pattern-link*,
17  management?
18 )>
19 <!ATTLIST pattern patternID CDATA #REQUIRED >
20 <!ELEMENT name (#PCDATA)>
21 <!ELEMENT alias (#PCDATA)>
22 <!ELEMENT illustration ANY>
23 <!ELEMENT problem (#PCDATA)>
24 <!ELEMENT context ANY>
25 <!ELEMENT forces ANY>
26 <!ELEMENT solution ANY>
27 <!ELEMENT synopsis (#PCDATA)>
28 <!ELEMENT diagram ANY>
29 <!ELEMENT evidence (example*, rationale?)>
30 <!ELEMENT example ANY>
31 <!ELEMENT rationale ANY>
32 <!ELEMENT confidence (#PCDATA)>
33 <!ELEMENT literature ANY>
34 <!ELEMENT implementation ANY>
35 <!ELEMENT related-patterns ANY>
36 <!ELEMENT pattern-link EMPTY>
37 <!ATTLIST pattern-link
38   type CDATA #REQUIRED
39   patternID CDATA #REQUIRED

```

```

40 collection CDATA #REQUIRED
41 label CDATA #REQUIRED
42 >
43 <!ELEMENT management (
44   author?,
45   credits?,
46   creation-date?,
47   last-modified?,
48   revision-number?
49 )>
50 <!ELEMENT author (#PCDATA)>
51 <!ELEMENT credits (#PCDATA)>
52 <!ELEMENT editor-comments (#PCDATA)>
53 <!ELEMENT creation-date (#PCDATA)>
54 <!ELEMENT last-modified (#PCDATA)>
55 <!ELEMENT revision-number (#PCDATA)>

```

**Auflistung A.1:** *Document Type Definition* der Pattern Language Markup Language nach Fincher [2003]

```

1 {
2   pattern_id: Number,
3   name: {
4     type: String ,
5     trim: true
6   },
7   alias: String ,
8   synopsis: String ,
9   confidence: {
10    type: Number,
11    min:0 ,
12    max:3
13  },
14  illustration: String ,
15  context: String ,
16  problem: {
17    type: String ,
18    trim: true
19  },
20  forces: String ,
21  solution: {
22    type: String ,
23    trim: true
24  },
25  rationale: String ,
26  diagram: String ,
27  example_description: String , // additional key
28  evidence: [{
29    rational: String ,
30    example: {

```

```

31     type: String ,
32     trim: true
33   }
34 },
35 literature: String ,
36 implementation: String ,
37 related_patterns: [{
38   type: {
39     type: String ,
40     enum: [ 'is-a', 'is-contained-by', 'contains' ]
41   },
42   pattern_id: Number,
43   collection: String ,
44   label: {
45     type: String ,
46     trim: true
47   },
48 },
49 ],
50 management: [{
51   author: String ,
52   credits: String ,
53   editor_comment: String ,
54   status: { // additional key
55     type: String ,
56     enum: [ 'proto-pattern', 'pattern', 'workshoped pattern' ]
57   },
58   creation_date: Date ,
59   last_modified: Date ,
60   revision_number: Number
61 }]]

```

**Auflistung A.2:** JSON-Adaption der PLML als *Mongoose Schema* Definition

# Kurzfassung der 45 Interaction Design Patterns für Videolernumgebungen

## ADD VIDEO

**Context:** The provided video learning environment contains a limited set of instructional videos. Other relevant videos might have been found or stored outside the platform.

**Problem:** Videos outside the provided learning environment are difficult to handle for both instructors and learners. First of all the learners need to know about the video. Second, the videos need to become accessible and remain accessible for all learners. Even if they are known and accessible the user will experience media breaks and might be destructured by unrelated content on the remote platforms.

**Solution:** Provide a learning environment where videos from different sources can be integrated in a consistent manner. Make sure learners get sufficient privileges to upload or embed resources in the system.

## ANNOTATED TIMELINE

**Context:** Enriched videos incorporate time-related contents like chapter marks or tags as well as further contents like pictures, websites or hyperlinks. Their presentation is determined by a temporal layout that contains the position and dimension of these events and marks.

**Problem:** The distribution of time-related and time-dependent annotations within the duration of a video is neither visible nor is it possible to navigate precisely to the corresponding positions in time. The learner should have a chance to develop its own mental model on how the interactive video works, and how its temporal events and marks relate to the playback time.

**Solution:** Indicate temporal events and scenes alongside the timeline by different icons, colours or on parallel tracks. Allow users to jump to these temporal positions.

## ANNOTATIONS

**Context:** A learning environment is not closed, but rather open for contributions made by teachers and students.

**Problem:** Compared to text or images, it is difficult to modify a video file. Videos are technically self-contained, which implies their use as a broadcast medium in terms of teaching and learning. The involved people are impeded from making contributions or additions that are directly related to the time-based information structure.

**Solution:** Make video extensible for the user through spatial and temporal annotations. Place them on top or beside the video.

## APPROPRIATE DELIVERY

**Context:** The file size as well as the experienced image quality of a video is determined by the product of the pixels per frame, their colour depth, and the frame rate. When delivering web video there is a trade-off between fast delivery and the quality of the audiovisual content. This is determined by the individual system and network capabilities of the user.

**Problem:** While users expect a fast delivery of a high-quality video they would be overwhelmed in finding the optimal way of content delivery.

**Solution:** Automatically deliver an appropriate video of maximum frame size and quality in a supported format that does not conflict with the possible data transfer capacity. Therefore provide several instances of the video that correlate with common screen resolutions, supported video formats, and network delivery capacities.

## ASSESSMENT

**Context:** A video consists of several content units that are build on each other. A key concept is explained in the video that students need to understand before continuing the video.

**Problem:** Learners tend to continue watching a video even if they did not understand the previous content section. With a lack of understanding, it becomes more and more difficult to comprehend the subsequent contents.

**Solution:** Pause the video after a key concept has been explained. Present a question as a little challenge that is suitable to determine how well the student understood the previous section. Since we focus on self-assessment it is important to provide immediate feedback including the correct solution. Neither make the submission of an answer mandatory nor define its result as a threshold for continuing the playback.

## BASIC CONTROLS

**Context:** An application is dedicated to deliver the content of a video to the user.

**Problem:** How to control the audiovisual presentation of a video?

**Solution:** Every video player needs to provide basic control elements regarding the playback (play / pause), temporal navigation (skip, timeline), display mode (normal view / full screen), and volume control.

## BRANCHING VIDEOS

**Context:** Learners differ in their interests, learning styles, and prior knowledge. In contrast to that an instructional video has been designed for an audience with particular interests and expected prior knowledge.

**Problem:** When watching a video learners are faced with different (content-driven) opportunities to leave the video. For instance, learners may want to close individual knowledge gaps or they may want to deepen a topic before or instead of continuing the playback. In this cases, they might not learn as much as they could if they continue the video. But if they would leave the video with an unclear goal in mind they will loose track towards the learning objectives.

**Solution:** Therefore, provide hyperlinks as branches on the designated learning path. Each branch is constructively aligned with your learning objectives. Offer branches of topic-related videos that address required prior knowledge, different interests of your audience, and desired topics for deepening learning.

## CLASSIFIED MARKS

**Context:** A large number of time-accurate annotations of the same type have been added to a video.

**Problem:** Viewers get confused by a large number of indistinguishable annotations and tend to miss the important ones.

**Solution:** Therefore, find a meaningful classification scheme that helps the user to distinguish and filter different annotation. Such a scheme could rely on the type (e.g. comment, hyperlink, quiz), the content (e.g. question, remark, and mistake for annotated comments), relevance or ratings (e.g. green, yellow, red).

## CLOSED CAPTIONS

**Context:** An instructional video is usually accompanied by spoken word in a certain language.

**Problem:** Audio information in a video can not be understood by audiences that are deaf, hard of hearing, or that have insufficient language competencies. These audiences are hardly able to participate in learning.

**Solution:** Support comprehension by providing closed captions of transcribed speech and other relevant audiovisual information. Let the user decide whether the subtitles should be visible or not. If they are visible they should be displayed inside the video frame.

## COMMENTS

**Context:** The video is delivered like broadcast media.

**Problem:** When watching an instructional video one might have questions, a remark or the will to start a discussion. In typical broadcast media, users can not express themselves while the content providers don't notice user feedback.

**Solution:** Let users anchor comments related to specific points in time. Also, enable re-comments to enable discussions. Do not restrict the form or style of the comments.

## DETAIL ON DEMAND

**Context:** Transfer of knowledge requires particular prior knowledge. The prior knowledge of individual learners almost always differs to some extent.

**Problem:** Within an instructional video you can neither expect the same prior knowledge from every learner nor can you explain every mentioned term or concept in detail to consider individual prior knowledge.

**Solution:** Provide necessary prior knowledge as details on demand. Therefore, use a hyperlink that refers to a segment of another video. As soon as the playback of the on-demand video ends the presentation of the original video should be continued.

## DIRECT AUTHORIZING

**Context:** The video delivered in the player can be extended by additional time-dependent and time-related contents. These annotations can be created, edited or removed.

**Problem:** The playback facilities are typically separated from the authoring tools. For adding or editing an annotation the user needs to switch from the player to the authoring interface in the back-end of the application. As a consequence, the task flow of watching and annotating is interrupted. Furthermore, it has a negative effect on the viewing experience. Finally, users hesitate to annotate content, because they do not want to loose focus on the current scene or topic.

**Solution:** Therefore, integrate the most important authoring tools directly in the user interface of the video player.



## FOLLOW REVISIONS

**Context:** Students are encouraged to extend the instructional video with additional information. They do this by working individually or in small groups. Furthermore, the augmented content and the video itself can be manipulated and changed.

**Problem:** It is difficult to determine which parts of a video or which video annotations have been added, modified or removed since the last visit. If users are not able to identify these changes they may miss important contributions from other users.

**Solution:** Implement comprehension of changes since the last visit. Provide a revision log that expresses different editing processes, whether they result in changes of visual, audio, and text content or the overall content structure (e.g. changes in the sequence of concatenated videos)

## FULL SCREEN

**Context:** A video is enriched by further time-dependent resources such as synchronized slides, OVERLAYS or ANNOTATIONS.

**Problem:** Browsers built-in full-screen presentation of the video dismisses time-related contents and navigational instruments. So the user cannot see all relevant contents and has only limited possibilities to control and navigate the video.

**Solution:** In full screen mode consider all contents that are related with the playback time of the video (e.g. synchronized slides). Also provide essential navigational elements such as TABLE OF CONTENT, HYPERLINKS or buttons to turn CLOSED CAPTIONS on or off.

## HYPERLINKS

**Context:** There are several videos collected in a video learning environment. Some contents relate on each other or contain implicit references to topics covered by another video or web resource.

**Problem:** Students often don't know about content-related relations between learning resources. It is difficult to make use of these relations because they are not easy to find. In case one might have identified a cross-relation between videos the relation remains implicit and not visible for other learners.

**Solution:** Therefore, make content relations explicit by linking segments in one video with another video or learning resource. Distinguish visually, whether a link points to another video within the same environment or to any other external resource. When referencing continuous media such as video or audio enable time-accurate link targets to address contents more precisely.

## JOURNALED NAVIGATION

**Context:** Browsing from one video to another leaves a trace that is not just resource-related but also related to the viewed scenes of each video.

**Problem:** Browsing between videos must be supported in both directions. While the forward directed links are commonly implemented, the way back is not considered.

**Solution:** Provide journaled navigation to go smoothly and time-accurately back and forth in the sequence of watched videos. In addition to the VIEWING HISTORY and the browser history users should be able to recall the starting point of the previously watched segments.

## LOADING INDICATOR

**Context:** Video browsing can be delayed by network latency, script execution and display problems.

**Problem:** User gets frustrated if the video player is not responding but remaining in an inexplicable state at start-up or during the playback.

**Solution:** Immediately indicate a delay of data transmission or presentation not just for the video, but also for other attached resources (e.g. slides, images or parts of the web page).

## LOOP

**Context:** The moving images of a video contain great visual details.

**Problem:** Understanding a visual presented process or animation sometimes requires the user repeat the presentation multiple times in order to grasp all relevant details. Jumping back on timeline after every run is very cumbersome, especially if the requested scene is only a few seconds long.

**Solution:** Therefore, enable the repetition of segments of a video. Inform the user when entering such a loop and provide a way to stop it.

## MEDIA FRAGMENTS

**Context:** Video has a very high resolution or includes multiple camera angles (e.g. a multi-angle video).

**Problem:** Analysis and discussions that only related to a very small part of the video require the user frequently to navigate and zoom.

**Solution:** Therefore, allow the user to extract a fragment of the video in order to work with this fragment. Provide handy tools to cut out parts of the video frame and to select a time range on the timeline.

## MULTI-TIMELINE EDITING

**Context:** A video has been augmented with different types of annotations. Some of the annotations overlap in their temporal extent.

**Problem:** Overlaps and clusters make it difficult to overview annotations that are displayed on a single timeline track. When managing these content annotations might be overlooked and ignored in the clutter.

**Solution:** For editing spread annotations on different timeline tracks. Preserve at least one track for each type of annotation.

## OBJECT TRACKING

**Context:** An annotated video has been augmented with OVERLAYS whose temporal and spatial position is related to a certain visual object in the video. This visual object can change its position over time.

**Problem:** The semantic relation of the annotation and its corresponding object in the video can not be ensured if they are not presented close to each other. Without this spatial context information an annotation can become meaningless.

**Solution:** Therefore, dynamically adjust the position of the annotation by analysing the movement of its corresponding object. Computer vision algorithm can be used for tracking the position of an object over time.

## OVERLAYS

**Context:** The instructor or students want to augment the video with further information. These additional contents can be synchronized with the video. The available space to present these contents is limited by the display size and the size of the video that is required to distinguish all necessary content.

**Problem:** Displaying synchronized additional information beside the video reduces the available display size of the video.

**Solution:** Present additional visual information in a time-dependent layer on top of the video. Make sure to place it closely to its corresponding visual contents. Avoid overlapping relevant visual information in the video.

## PLAYBACK SPEED

**Context:** The presentation of video contents can result in the two extremes of a cognitive load. On the one hand, overload can be caused by fast scene changes or volatile details in visual dynamic processes. Mental underload, on the other hand, is related to almost unchanging images or irrelevant content.

**Problem:** The density of presented information in a video differs from the expected level of detail, sometimes it is perceived as too slow, sometimes as too fast.

**Solution:** Integrate a mechanism to change the speed of presentation from slow motion to normal speed to time-lapse and vice versa.

## PLAYLIST

**Context:** Videos in a large collection are filed in a predefined category system.

**Problem:** When browsing a large collection of videos, users need to organize resources by their own needs. Selected, previously watched resources should not need to be searched over and over again.

**Solution:** Provide learners a playlist, where they can collect and store subsets of a video collection for later use. These personal subsets can be freely extended or reduced by videos that are part of the learning environment.

## RELATED VIDEOS

**Context:** A set of videos that strongly build on each other in a sequential manner should be rather interlinked with each other or concatenated in the way of SEQUENTIAL MEDIA. Hence additional learning resources are not intended to provide a linear learning path. The content of a video bears many implicit ties to other learning resources. Learners can be motivated to follow these ties and consume resources that are part of the learning environment.

**Problem:** The desire to watch another video awakes during or after a video presentation. The viewer might be looking for another video during the playback if the one presented does not fulfil his/her needs. After completion, the viewer would like to continue with a subsequent or deepening lecture. In both cases, the learner should not leave the learning environment if he/she could proceed the learning with an appropriate resource.

**Solution:** Always offer links to additional learning resources that fosters continued learning activities. Provide these links after one video has ended. In addition to that display a set of related videos beside the current video. Consider resources that are semantically or didactically related to the current video.

## REMIX

**Context:** Producing a video from scratch takes a lot of time. The same is true for video annotations such as TABLE OF CONTENT, **Hyperlinks** or **Assessment**. The video and its annotations can be modified or extended by other users.

**Problem:** To create a new video users cannot make use of existing parts of a video or video annotations.

**Solution:** Therefore, allow users to take an instance of a video including all its annotations as a starting point for their own version or remix of the original content. Make it possible to integrate or connect artefacts from different sources.

## SEARCH

**Context:** A video can be seen as a continuous stream of audible and visual information. This information can not be scanned as quickly as a piece of text. The longer a video the harder it gets especially to overview the verbal content.

**Problem:** Verbal information stored in a video is not accessible at a stroke. Increasing the playback rate or listening all speech takes too much time. Besides that, one may also have to look for written words that are part of the video and its synchronized media streams. In consequence, the verbal content related to a video can not be searched directly.

**Solution:** Enable a search within all available sources that are related to the video. Consider video related meta data and all available media streams including audio. Take also information into account that needs to be extracted from annotated contents such as synchronized slides, images, tags, or links. Provide a link from each search result back to its corresponding artefact at a given playback time.

## SEQUENTIAL MEDIA

**Context:** There is a set of videos that should be watched in a certain order.

**Problem:** If the video would be watched in a different order it would become difficult to comprehend them. Important prior knowledge (e.g. about terms, concepts, theories) could be missing or the sequence of an explanation or a process order would be messed up.

**Solution:** Therefore, sequence the videos in a fixed order and make the order visible to the user. Provide a menu or enumerated list to emphasize the sequence of videos. Foster navigation between adjacent videos through direct links. Automatically load the subsequent video when a video ends.

## SHORTCUT COMMANDS

**Context:** Video players incorporate different designs and provide different user interfaces.

**Problem:** Graphical user interfaces are not accessible for the blind without a screen reader. These interfaces can also become frustrating for users who are not impaired in their vision, especially for frequently executed tasks.

**Solution:** Enable keyboard shortcuts and touch gestures for the most frequent user interactions. Specifically, define those shortcuts that most of the video players have in common.

## **SIMULTANEOUS MEDIA**

**Context:** Different types and formats of instructional media are provided to enable learning, whereas the given video contains the most important information.

**Problem:** A subject matter is not fully covered by the video, but provided with separate learning resources that are indispensable for learning. Despite the strong relations between the video and the other resources, they are stored and presented separately.

**Solution:** Synchronize corresponding contents with segments of the video. Present them side by side to the video.

## **SKIP BACK**

**Context:** The visual presentation is accompanied by a narration. Or the visual contents has volatile details.

**Problem:** The user can have difficulties to understand currently spoken words or to recognize volatile visual details. In terms of understanding audio, one might not be familiar with the language, the used vocabulary the level of difficulty. Regarding volatile visual content, an object might disappear after a very short time or change their gestalt or look.

**Solution:** Provide a short-cut button to jump back a few seconds back in time in order to repeat the last sentence or the previous scene of the video image.

## **SYNCHRONIZED MAP**

**Context:** The content of a video contains many geographic or location-related references. These details are not directly visualized in the video (e.g as a map or an arial image).

**Problem:** Learners who can not integrate places or regions mentioned in the video will have difficulties to comprehend the contents that are related to these locations.

**Solution:** Show an interactive map beside the video that provides the geographic context. Synchronize the presentation of locations, routes, and areas, but do not restrict the user to navigate the map.

## **TABLE OF CONTENT**

**Context:** The video is longer then 20 Minutes. Its content stretches across several sections.

**Problem:** Content units of a video can be neither overviewed nor accessed precisely. So the user wastes time for searching and navigating.

**Solution:** Provide a navigable table of content next to the video that provides a brief overview of the video content. Enable time accurate access to the corresponding content sections. Highlight the currently watched section in the table of content.

## TEMPORAL BOOKMARKS

**Context:** In a video learning environment links are used to reference a video to be loaded by a player. These links can be easily copied, saved or shares. Particularly they are used to bookmark the video or for setting a link somewhere else in the web.

**Problem:** Time-related events or periods in the video are not part of a URL. If you are sharing a URL you cannot point to a certain moment or period in the video. This information needs to be transferred separately. Often it is missing or only verbally described. The one who would like to watch a particular scene has to open the link and navigate manually to the desired scene.

**Solution:** Implement media fragments like time stamps, periods or section names as part of the URL. When opening a URL that includes these fragment identifiers parse them and navigate the video player to the given temporal position.

## TEMPORAL TAGS

**Context:** Many videos have been collected in a video learning environment. They were filed into categories and subcategories. In addition, the video files were annotated with keywords (tags).

**Problem:** Tags that are related to an entire resource do not tell the user what parts of the video cover the related contents. The user has to navigate and search to find the corresponding time stamps or periods. It is also unclear how often a tag occurs in the video.

**Solution:** Therefore, provide tags that correspond to a point or period of time. Combine tags in a separate view that belong to the current video. Offer another view that integrates temporal tags from multiple videos (e.g. all videos in a certain category). Facilitate browsing by linking each tag to its (multiple) time-accurate link target(s).

## TRANSCRIPT

**Context:** An instructional video is usually accompanied by spoken language.

**Problem:** Auditory information of a video is volatile and can be neither accessed or skimmed at the users own pace.

**Solution:** Provide a transcript of all auditory information as a text equivalent beside the video. Make this transcript navigable and highlight paragraphs that correspond to the current playback position.

## USER NOTES

**Context:** Comprehensive knowledge is delivered by video.

**Problem:** Taking notes while watching a video can be cumbersome if the speaker talks faster than the user is writing down things. Notes on a sheet of paper or in a text document make it difficult to link back to the video. Because of the missing relations notes remain separate from the video.

**Solution:** Enable the user to take personal notes during the video playback. Relate each section of the notes to a segment of the video. Make these relations navigable in both directions - from a video segment to a section in the notes and vice versa.

## USER RATINGS

**Context:** User have access to a comprehensive collection of videos. These videos can be organized by categories and tags.

**Problem:** Users are overwhelmed of similar videos. Differences in quality or relevance can not be recognized at a glance. They cannot estimate which videos are worth to watch.

**Solution:** Therefore, let users rate the videos regarding their quality or relevance. Show the rating results whenever users will have the choice to select one of many video, e.g. in a list of search results or category contents.

## USER TRACES

**Context:** In asynchronous communication learners tend to miss each other when watching a video. So they are not aware of each other and cannot relate to their progress in learning.

**Problem:** Learners do not know what contents their peers have watched or skipped. Both instructors and learners are interested in what parts of the video had been already watched and processed by a group.

**Solution:** Visualize the frequency of perception on the timeline so that one can estimate which parts they have been watching often, rarely or not at all. Provide these user traces to peers and the instructor.



## VIDEO MANAGER

**Context:** The video learning environment contains a collection of videos that might be organized by categories and tags.

**Problem:** User can select videos from a list or menu. Because of a lack of sufficient information about the video content the user needs to open the videos one by one in order to find the desired one.

**Solution:** Support browsing and searching by providing a meaningful overview of any set or subset of the videos in the system. Include necessary meta data (e.g. title, author, tags, description, duration, ratings) as well as a visual preview. Provide a direct link to the video player in order to start the playback.

## VIDEO MANIPULATION

**Context:** An uploaded video file has some failures, e.g. the image is too dark, it is wrong orientated or the first three seconds need to be trimmed.

**Problem:** If a video has been uploaded the file has to be used as is. For editing the file needs to be (1) downloaded, (2) opened in a video editing software, (3) edited and re-encoded, and (4) finally uploaded to the system again. For applying only minor changes this process is too time-consuming and reduces the quality because of the re-encoding.

**Solution:** Enable basic editing tools that only manipulate the presentation and leave the file untouched. For example, one could adjust image properties like brightness, contrast, and colours by using filters. One might also need to change the orientation from landscape to portrait. Unwanted scenes can be excluded from playback instead of trimming or cutting the file.

## VIEWING HISTORY

**Context:** Learners have access to a collection of videos that they want or need to watch.

**Problem:** The learner cannot exactly remember which videos they have already watched. Furthermore, they hardly remember what parts they have seen so far.

**Solution:** Offer a viewing history that lists the previously watched videos. Make sure the user can estimate to what extent a video has been watched already.

## VISUAL HIGHLIGHTING

**Context:** A narrated video includes multiple visual aspects or details that are relevant to understand the content.

**Problem:** Learners are not aware of a visually presented object that is referred to by the narration or online discussion.

**Solution:** Highlight visual objects or details inside the video as long as they are necessary to understand the oral presentation or content-related communication.

## VISUAL SUMMARY

**Context:** Dynamic visualizations such as videos are presented little by little. Despite an increased PLAYBACK SPEED it can not be scanned as quickly as a piece of text. The longer a video the harder it gets to overview the visual content.

**Problem:** Visual information stored in a video is not accessible at a stroke. Browsing along the timeline can be cumbersome the longer the video and the more similar the scenes are. Neither visual content can be overviewed nor certain visual anchors are searchable.

**Solution:** Therefore, provide an overview of the visual contents that summarizes the information inside the player and in the VIDEO MANAGER. Represent the summary in the VIDEO MANAGER as generic still images or flicker-book-like animations to overview and search visual cues. In addition, provide a scrubber preview on the timeline of the player. When hovering a point on the timeline the corresponding preview image should be presented.

## ZOOM

**Context:** A video or its synchronized contents contain visual details that are essential for comprehension.

**Problem:** Users miss essential visual details that are presented too small.

**Solution:** Therefore, it is advisable to let the user zoom in and out all time-dependent contents including the video itself.

## Beispiele für Scripts/CSCL-Scripts

Der Ablauf und die groben Rahmenbedingungen der in Tab. A.18 angeführten Scripts und CSCL-Scripts finden im Folgenden eine Erläuterung.

**Tabelle A.18:** Beispiele für Scripts und CSCL-Scripts (●). Scripts, die sowohl mit Hilfe des Computers, als auch ohne durchgeführt werden können sind mit ○ gekennzeichnet.

Bezeichnung	Quelle	CSCL
ArgueGraph	z.B. Jermann und Dillenbourg [1999]	○
Dyadic Essay Confrontation	Millis und Cottell [1997]	–
ICE Script	Dillenbourg [2006]	●
Jigsaw	Aronson et al. [1978]	○
MagicBook	Dillenbourg [2002]	–
Maze Script	Miao et al. [2005]	●
MURDER Script	O'Donnell und Dansereau [1993]	–
Peer Editing	Johnson et al. [2006]; Millis und Cottell [1997]	
PeerQuest	Millis und Cottell [1997]	○
Phase-X	Engeli [2001]	●
Reciprocal Teaching	Palincsar und Brown [1984]	–
Social Script	Weinberger et al. [2005]	–
UniverSante Script	Berger et al. [2001]	●

### ArgueGraph

Das Script wurde erstmals von Jermann und Dillenbourg [1999] beschrieben und untersucht.

Ziel des Scripts ist es, einen Konflikt zwischen den Studierenden zu erzeugen, den sie anschließend selbst auflösen sollen. In dem Script sind sowohl kollaborative, als individuelle Lernformen vorgesehen. Es lässt sich sowohl in Präsenz ohne die Unterstützung von Computern, als auch in CSCL-Lernumgebungen umsetzen. Siehe auch: [http://manyscripts.epfl.ch/available\\_scripts](http://manyscripts.epfl.ch/available_scripts)

Möglicher Ablauf:

1. Alle Lernenden beantworten Multiple Choice Fragen in einem (Online-)Quiz. Die Fragen sind so gewählt, dass sie Meinungen der Teilnehmenden polarisieren, anstatt Konsens herbeizuführen. Für jede Antwort wird eine Begründung erwartet.
2. Der Lehrende bildet Paar-Gruppe aus denjenigen Lernenden, deren Antworten sich im Quiz möglichst stark widersprochen haben.
3. Die Paare bearbeiten das Quiz erneut und sollen sich auf eine gemeinsame Antwort einigen und diese begründen.

## Dyadic Essay Confrontation

Siehe Millis und Cottell [1997]

Ablauf:

1. Die Lernenden erhalten die Aufgabe einen Text zu lesen und sollen dazu eine Frage formulieren. Die Frage kann sich auch auf bisherige Lerninhalte beziehen. Die Frage wird auf einem Blatt Papier festgehalten. auf einem weiteren Blatt schreiben sie die Musterantwort.
2. In der nächsten Phase bilden die Lernende Paare (Dyaden), in denen die Lernenden ihre Fragen austauschen und auf einer Seite beantworten.
3. Innerhalb der Gruppen werden die Antworten und die Musterantwort zwischen den Gruppenmitgliedern ausgetauscht. Sie lesen nun beide Texte, Antwort und Musterantwort.
4. Abschließend vergleichen und diskutieren sie ihre Lösungen in der Gruppe.

## Ice Script

Das *Ice Script* ist Bestandteil der scriptbasierten CSCL-Umgebung *ManyScripts*. Es wird im wesentlichen ein Peer Assessment umgesetzt. Siehe: [http://manyscripts.epfl.ch/available\\_scripts](http://manyscripts.epfl.ch/available_scripts).

Ablauf:

1. Jeder Teilnehmer erledigt eine vom Lehrenden vorgegebene Aufgabe. Die Aufgabenlösung wird in Textfelder eingegeben oder als Bild oder Datei in die Lernumgebung eingebunden.
2. Teilnehmer geben einer bestimmten Anzahl ihrer Peers Rückmeldungen zu ihren Aufgabenlösungen.
3. Jeder Teilnehmer liest die Rückmeldung zu seiner Aufgabenlösung und überarbeitet die eigene Lösung.
4. Jeder Teilnehmer bestimmt nun die beste aller Lösungen. Anhand dieser Auswahl wird eine Rangreihe der besten Lösungen erstellt und den Teilnehmern präsentiert.

## Jigsaw

Das Script geht zurück auf Aronson et al. [1978] und wurde bereits Anfang der 1970er Jahre in amerikanischen Schulen angewandt.

Ablauf: Aus einer Kohorte werden heterogene Gruppe von fünf bis sechs Personen gebildet. Zudem definiert der Lehrende eine Aufgabe, die sich in ebenso viele Segmente aufteilen lässt, wie es Mitglieder in einer Gruppe gibt. Jedes Gruppenmitglied soll eine Teilaufgabe bearbeiten. Dazu bekommt es exklusiven Zugang zu den dafür notwendigen Informationen. Somit kann kein Teilnehmer die Gesamtaufgabe allein bewältigen. Würden die Informationen einfach an jeden in der Gruppe durchgereicht, müsste sich jeder zusätzlich in den entsprechenden Teilbereich einarbeiten, um zum Experten der Subdomäne zu werden. Jedes Gruppenmitglied bekommt ausreichend Zeit seine Teilaufgaben zu bearbeiten und selbige auch noch einmal zu überarbeiten. Anschließend werden die Gruppen in der Art aufgeteilt, dass sich all diejenigen in Gruppen zusammenfinden, die die gleiche Teilaufgaben bearbeitet haben. Innerhalb dieser Ex-

pertengruppen tauschen sich die Lernenden aus und diskutieren zentrale Ergebnisse. Daraus ergibt sich die Gelegenheit die individuellen Ergebnisse noch einmal zu überarbeiten und zu verbessern. Anschließend wechseln die Teilnehmer wieder zurück in ihre ursprüngliche Gruppe und stellen sich gegenseitig die Arbeitsergebnisse der Teilaufgaben vor.

## MagicBook Script

Das MagicBook Script ist u.a. bei Dillenbourg [2002] nachzulesen.

Ablauf:

1. Der Instruktor schreibt den Beginn einer Geschichte an die Tafel.
2. Alle Lernenden lesen den Text.
3. Die Lernenden schreiben individuell einen Entwurf für das zweite Kapitel der Geschichte.
4. Alle Lernenden lesen die Entwürfe ihrer Peers und wählen ihren Favoriten aus.
5. Der gewählte Entwurf wird zum zweiten Kapitel.
6. Der Vorgang zwischen Schritt 3 und 5 iteriert beliebig oft.
7. Für das letzte Kapitel sind die Lernenden selbst aufgefordert ein Ende der Geschichte zu formulieren.

## Maze Script

Das *Maze Script* ist bei Miao et al. [2005] nachzulesen.

Für die beispielhafte Aufgabe, eine Labyrinth und einen Labyrinthroboter zu entwickeln, gestaltet sich der Ablauf des Scripts wie folgt:

1. 24 Lernende teilen sich in 6 Gruppen à 4 Personen auf.
2. Je zwei Personen in einer Gruppe entwickeln Strategien für den Roboter („strategy developer“) und den Aufbau des Labyrinths („maze builders“). Die Ergebnisse werden auf einem Server bzw. in der Lernumgebung abgelegt.
3. Ein gewählter Gruppensprecher präsentiert die Ergebnisse und alle Gruppenergebnisse werden den übrigen Gruppen zugänglich gemacht.
4. *strategy developer* und *maze builder* tauschen ihre Rollen und wetteifern um die besten Strategien bzw. Labyrinth bezüglich der anderen Gruppen.
5. Die Gewinnergruppe stellt ihre Ergebnisse vor.

## MURDER Script

Das MURDER Script von O'Donnell und Dansereau [1993], auch allgemein als *Scripted Collaboration* bekannt, ist für zwei Personen ausgelegt.

1. Mood: Einstimmung auf die Bearbeitung eines Textes.
2. Understand: individuelle Lektüre des ersten Textabschnittes, festhalten von Kerngedanken oder Fakten.
3. Repeat: Person A wiederholt den Inhalt aus dem Gedächtnis.
4. Detect: Person B gibt A Feedback zu dessen Zusammenfassung und deckt Lücken und Widersprüche auf.

5. Elaborate: gemeinsames Erarbeiten des Textes. Verknüpfen von Erfahrungen und Vorwissen.
6. Review

Das Script wird für mehrere Textabschnitte wiederholt, wobei sich die Lernenden in Phase 3 und 4 in den Rollen abwechseln. Siehe auch weitere Variationen bei Mandl und Ertl [2004]

## Peer Editing

Das Script ist nachzulesen bei Millis und Cottell [1997] und Johnson et al. [2006].

Ablauf:

1. Lernende in einem Kurs haben für eine Hausarbeit o.ä. einen Text verfasst.
2. Einige Tage vor dem Abgabetermin übergeben sie den Textentwurf einem anderen, vorher festgelegten Kursteilnehmer. Jeder Teilnehmer begutachtet nun einen Entwurf eines Peers durch kritische Anmerkungen und Kommentare im Text. Falls es sich um längere Texte handeln, gilt es im Vorhinein einen Schwerpunkt der Begutachtung festzulegen (z.B. bestimmte Abschnitte oder Fragestellungen).
3. Ausreichend viele Tage vor dem finalen Abgabetermin erhalten die Teilnehmer die begutachteten Arbeiten zurück, so dass sie die Anmerkungen noch rechtzeitig einarbeiten können.

## PeerQuest

Siehe Millis und Cottell [1997] sowie folgende Quelle für eine Zusammenfassung unter dem Titel „Guided Reciprocal Peer Questioning“ ; siehe <http://archive.wceruw.org/cl1/cl/doingcl/peerqst.htm> (abgerufen am 8.02.2016).

Ausgangspunkt dieses Scripts können Lernende sein, die sich mit einem neuen und anspruchsvollen Lerngegenstand konfrontiert sehen, der sie so weit überfordert, dass sie keine adäquaten Fragen finden, um ein besseres Verständnis zu erlangen. Die Fähigkeit zur Formulierung von Fragen wird deshalb zunächst durch das Angebot von Satzanfängen von Fragesätzen unterstützt. Typische Satzanfänge lauten: Welches sind die Vor- und Nachteile von ... ? Welches andere Beispiel kann man für ... anführen? Erkläre, warum ... ?

Ein beispielhafter Ablauf könnte wie folgt aussehen:

1. Die Lernenden lesen einen Text, verfolgen eine Präsentation oder betrachten ein Video.
2. Jeder vervollständigt fünf Fragen mit Hilfe der gegebenen Satzanfänge.
3. Die Lernenden schließen sich in Kleingruppen von 3 bis 5 Personen zusammen und diskutieren mögliche Antworten auf die von ihnen formulierten Fragen.

## Phase-X

Das Phase-X Script wurde durch Engeli [2001] vorgestellt<sup>23</sup>

Eine Kohorte von Lernenden wird in Gruppen aufgeteilt. Alle Gruppen erhalten die Aufgabe ein Projekt zu bearbeiten, d.h. alle Gruppen haben das gleiche Ziel. Das Projekt ist in verschiedene Phasen unterteilt. Die Arbeitsergebnisse einer jeden Phase werden in einem gemeinsamen Ordner (*shared space*) zusammengetragen. Nach Abschluss einer Phase können die anderen Gruppen die Ergebnisse der zurückliegenden Phase einsehen und sie in der aktuellen Phase verwenden.

Beispiel: In einem Praktikum im Software Engineering besteht das Ziel darin, in einer Gruppe von drei Personen eine aus vier Programmmodulen bestehende Anwendung zu entwickeln. Je Modul ist eine Phase in dem Projekt vorgesehen. Nach Abschluss der Phase erhalten die Teilnehmer Einblick in die Lösungen der anderen Gruppen und können selbst entscheiden, ob sie Anregungen zur Verbesserung ihres Moduls umsetzen oder gleich mit einem Modul einer anderen Gruppe weiterarbeiten.

## Reciprocal Teaching

Das Script haben Palincsar und Brown [1984] beschrieben. Die Methode wird manchmal auch als *Pair Reading* bezeichnet. Erfolgreich eingesetzt wurde dieses Script vor allem bei Lernenden mit Schwierigkeiten beim Lesen und Verstehen von Texten. Dazu bilden jeweils zwei Personen eine Gruppe. Eine Person liest einen Absatz vor, zu welchen die andere Person Verständnisfragen stellen kann. Bei jedem neuen Absatz wechseln die Beteiligten ihre Rolle zwischen Vorleser und Fragesteller.

## Social Script

Dieses Script wurde von Weinberger et al. [2005] beschrieben und untersucht.

Ablauf: Eine Gruppe von drei Lernenden bearbeiten gleichzeitig je eine von drei Lernressourcen (z.B. Fallstudien oder Buchkapitel). Jeder schreibt eine Zusammenfassung. Anschließend kritisiert jeder die Zusammenfassungen der anderen beiden Gruppenmitglieder in einer konstruktiven Weise. Als Unterstützung der Analyse und Kritik werden bestimmte Satzanfänge angeboten, welche die Lernenden durch Fakten vervollständigen können.

## UniverSanté Script

Dieses Script gestaltet ein sehr komplexes Lehr-Lern-Szenario, welches eine Kollaboration im virtuellen Raum erfordert, jedoch nicht ohne Präsenztreffen auskommt. Siehe Berger et al. [2001] sowie auch Kobbe et al. [2007] und Dillenbourg [2002].

---

<sup>23</sup>Siehe auch <http://old.arch.ethz.ch/space/space.arch.ethz.ch/indexd88ed88e.html?choice=10> (abgerufen am 27.01.2016).

Ablauf:

1. Teilnehmer aus verschiedenen Nationen lesen und diskutieren eine Problemstellung in gemischten Gruppen.
2. Je Nationalität wird eine Gruppe gebildet. Die Gruppenmitglieder informieren ihre Gruppe über die zuvor bearbeitete Problemstellung und die entwickelte Lösung. Die Gruppe erstellt dazu gemeinsam ein Merkblatt („fact sheet“).
3. Nun werden Gruppen aus Teilnehmern gebildet, die ein ähnliches Problem bearbeitet haben. Sie vergleichen die Lösungen der aus den Ländergruppen hervorgegangenen Merkblätter und destillieren die wesentlichen Aspekte zu einem überarbeiteten Merkblatt.
4. Die Teilnehmer kehren in die Gruppe je Nationalität zurück und stellen das Merkblatt vor.
5. Alle Teilnehmer kehren in die Gruppe aus Phase 1 zurück und bearbeiten das Problem.



## VI-TWO Codebeispiele

### Instantiierung

Im Konstruktor der in Auflistung 34 dargestellten exemplarischen Klasse Vi2.Example (Z. 3) wird die Datenbasis als JSON geladen, um anschließend das Framework mit dem Observer (Z. 16-21) zu initiieren. In Zeile 22 wird dann ein Video aus der Datenbasis geladen. Im Anschluss werden die Widgets für den Video Manager (Z. 24-26), die Historie der betrachteten Video (Z. 28/29) und die Anpassung der Wiedergabegeschwindigkeit (Z. 31/32) erzeugt und im Observer initiiert.

```

1 Vi2.Example = function() {
2     vi2 = this;
3     vi2.dom = "#vi2";
4     vi2.templatePath = "views/";
5
6     var files = [
7         {path: 'http://www.example.com/data.json', storage: 'json_data'}
8     ];
9     vi2.utils = new Vi2_Utils();
10    vi2.db = new Vi2.DataBase( {path: '/', files: files }, this, 'init');
11 };
12
13 Vi2.Example.prototype = {
14     init : function(){
15         vi2.observer = new Vi2.Observer({
16             selector: "#seq",
17             videoWidth: "400px",
18             videoHeight: "800px"
19         });
20         vi2.observer.init(0);
21         vi2.observer.setCurrentStream('video1');
22
23         var videoManager = new Vi2.VideoManager();
24         vi2.observer.addWidget( videoManager );
25         videoManager.init();
26
27         var viewingHistory = new Vi2.ViewingHistory();
28         vi2.observer.addWidget( viewingHistory );
29
30         var playbackSpeed = new Vi2.PlaybackSpeed();
31         vi2.observer.addWidget( playbackSpeed );
32     }
33 };

```

**Auflistung A.3:** Exemplarische Klasse zur Initialisierung von VI-TWO

## Widget HelloWorldVideoAnnotation

Das in Auflistung 38 dargestellte exemplarische Widget Vi2.HelloWorldVideoAnnotation dient dazu Annotationen des gleichnamigen Types (Z. 6) als ein Ereignis während der Videowiedergabe auszulösen (Z. 37) und zu terminieren (Z. 41). Die zeitliche Verteilung der Ereignisse wird gemäß der Optionen (Z. 8-10) auch auf der Zeitleiste dargestellt (Z. 28-34).

```

1 Vi2.HelloWorldVideoAnnotation = $.inherit( Vi2.Annotation , {
2   __constructor : function(options) {
3     this.options = $.extend(this.options , options);
4   },
5
6   name : 'helloWorldVideoAnnotation' ,
7   type : 'annotation' ,
8   options : {
9     hasTimelineMarker: true ,
10  },
11  elements : [] ,
12
13  init : function(annotations){
14    var _this = this;
15    events = annotations.map(function(val){
16      if(val.type === _this.name ){
17        return { name: val.title , time:[val.t1] };
18      }
19    });
20
21    if( this.options.hasTimelineMarker ){
22      vi2.observer.player.timeline.addTimelineMarkers(
23        'helloWorldVideoAnnotation' ,
24        events ,
25        this.options.timelineSelector
26      );
27    }
28  },
29
30  begin : function(e, id , obj){
31    console.log( 'hello world' );
32  },
33
34  end : function(e, id , obj){
35    console.log( 'good bye' );
36  },
37 });

```

**Auflistung A.4:** Exemplarisches Widget für VI-TWO

# Spezifikation von Hypervideo-Dokumenten in VI-TWO

## HTML

**Tabelle A.19:** Spezifikation benutzerdefinierter HTML data-Attribute, die für den Anwender nicht sichtbar im Quellcode eines Dokuments verborgen sind. Für jede Art von Video-Annotation sind verpflichtende (●) und optionale (○) Attribute vorgesehen.

Attribut	Datentyp	HTML Custome Elements											
		vi2-video	vi2-chapter	vi2-tag	vi2-link	vi2-link-cycle	vi2-highlight	vi2-media	vi2-map-point	vi2-caption	vi2-transcript	vi2-comment	vi2-note
Metadaten													
data-author	String	○	○	○	○	○	○	○	○	○	○	○	●
data-created	Date	●	●	●	●	●	●	●	●	●	●	●	●
data-updated	Date	●	●	●	●	●	●	●	●	●	●	●	●
Inhalt													
data-label	String	●	●	●	●	●	○	○	○				
data-target	String				●	●	●		○				
data-content	String								○	●	●	●	●
data-src	String	●				●		●					
Zeitparameter													
data-start	Number / String		●	●	●	●	●	○	●	●	●	●	●
data-duration	Number	○	○	○	●	●	●	○	●	●	●	●	
data-target-start	Number / String				○	●							
data-target-duration	Number				○	●							
Raumparameter													
data-x	Number				●	●	●		●			○	
data-y	Number				●	●	●		●			○	
data-width	Number	○			○	●	●	○					
data-height	Number	○			○	●	●	○					

## Vi-Wiki markup

Im Folgenden wird ein Auszug aus der Benutzeranleitung des Vi-Wiki wiedergegeben und die Verwendung der erweiterten *wiki markup* erklärt. Siehe auch Seidel [2011a].

### 1. Getting started

Vi-Wiki source is written inside the hypervideo tag:

```
1 <hypervideo>
2 ...
3 </hypervideo>
```

### 2. Include Video

Videos can be assigned in the same way as images in the wiki markup of MediaWiki. So its one line of markup code to embed a video which fills out the text width and which is playing in full length. Its spatial height results from the display ratio of the media element.

```
1 [[ Video: <url> | <unique-id> | \#<start-time> | <duration> |<width> ]]
```

- url (required): Resource location in the WWW or within the wiki
- unique-id (required): The unique identifier is required for referencing the video. It can be an arbitrary String but without any blanks.
- start-time: Temporal offset when playback starts. It has the suffix # similar to references on HTML labels. The time can be defined in seconds or in decimal format (hh:mm:ss). Default: 0 (seconds).
- duration: Timespan for how long the video will play from its start time. The duration can be defined in seconds or in decimal format (hh:mm:ss). Default: 0 (seconds).
- width: Horizontal dimension of the video in pixel (px) or percent (%). Default: 100 %.

### 3. Add Hyperlinks

The notation for hyperlinks follows the markup for video elements. In general links can overlap in space and time. The link that appears last overlaps the previous ones. It is obvious that there is no syntactical distinction between different link types necessary. The parser determines internal, external (page on different domain) or non existing hyperlinks and provides a particular link icon during playback.

```
1 [[ <link-target> <link-label> ] \#<start time>
2 | <duration> | <x-position> | <y-position> ]
```

- link-target (required): URL of an external resource or page within the wiki or unique-id of a video. The letter can also refer to another temporal position inside the same video.
- link-label: Label of the link anchor. The anchor consists of an icon and a text label. The icon suggests whether the link target is within the same video, on another wiki page or outside the site. Default: (blank).
- start-time: Temporal position when the link will appear. It has the suffix # similar to references on HTML labels. The time can be defined in seconds or in decimal format (hh:mm:ss). Default: 0 (seconds).
- duration: Timespan for which the link will be visible after its appearance. The duration can be defined in seconds or in decimal format (hh:mm:ss). Default: 10 (seconds).

- x-position: Spatial position on x-axis where the link anchor appears within the video frame. The position relates the top left corner of the link anchor. The position is defined in percent (%). Default: 50 %.
- y-position: Spatial position on y-axis where the link anchor appears within the video frame. The position relates the top left corner of the link anchor. The position is defined in percent (%). Default: 50 %.

#### 4. Concluding example

```

1 <hypervideo>
2 [[ Video: http://example.com/video.webm myVideo | 200px ]]
3 [[ Main Page | WikiSym ] \#10 | 20 | 26% | 26% ]
4 [[ http://www.wikisym.org/ Click me ] \#80 | 120 | 50% | 50% ]
5 [[ Video: demo.ogv anotherVideo | 200px ]]
6 [[ Some Page | see here ] \#10 | 20 | 26% | 26% ]
7 </hypervideo>

```

## Relevante Publikationen im Kontext dieser Arbeit

### Schriften (Auswahl):

- Seidel, N. (2011). Enable Wikis for Seamless Hypervideo Integration. In Proceedings of the 29th Annual European Conference on Cognitive Ergonomics (S. 251–252). Rostock, Germany: ACM. doi: 10.1145/2074712.2074765
- Seidel, N. (2012a). Collaborative hypervideo editing using MediaWiki. In Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration (S. 22:1–22:4). Linz: ACM. doi: 10.1145/2462932.2462962
- Seidel, N. (2012b). E-Learning-Modul on Integrated Water Resources Management: Konzepte und Werkzeuge für die Realisierung einer hypervideo-basierten Lernumgebung. In J. Desel, J. M. Haake, & C. Spannagel (Hrsg.), DeLFI 2012 – Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik. (S. 291–302). Bonn: Gesellschaft für Informatik.
- Seidel, N. (2013). Peer Assessment und Peer Annotation mit Hilfe eines videobasierten CSCL-Scripts. In A. Breiter & C. Rensing (Hrsg.), DeLFI 2013 – Die 11. e-Learning Fachtagung Informatik der Gesellschaft für Informatik. (S. 83–94). Bonn: Gesellschaft für Informatik. **(nominiert für den Best Paper Award)**
- Seidel, N. (2014a). Analyse von Nutzeraktivitäten in linearen und nicht-linearen Lernvideos. Zeitschrift für Hochschulentwicklung - Videos in der (Hochschul-)Lehre, 9(3), S. 164–186. <http://www.zfhe.at/index.php/zfhe/article/view/687> (abgerufen am 18.07.2016)
- Seidel, N. (2014b). Ein Untersuchungsdesign zum Vergleich von offener und Script-basierter Kollaboration beim Lernen mit Videos. In DeLFI 2014 – Die 12. e-Learning Fachtagung Informatik der Gesellschaft für Informatik. (S. 297–300).
- Seidel, N., & Jödicke, C. (2014). Aufgabentypen und Einsatzszenarien für ein E-Assessment in Lernvideos. In K. Herring & J. Kawalek (Hrsg.), WEL'14 - Workshop on eLearning (S. 83–92). Görlitz.
- Seidel, N. (2014c). Interaction design patterns for design and development of video learning environments. In Proceedings of the 19th European Conference on Pattern Languages of Programs (S. 20:1–20:12). New York, NY, USA: ACM. doi: 10.1145/2721956.2721981
- Seidel, N. (2014d). Interaction design patterns for interactive video players in video-based learning environments. In Proceedings of the 19th European Conference on Pattern Languages of Programs (S. 19:1–19:14). New York, NY, USA: ACM. doi: 10.1145/2721956.2721980
- Seidel, N. (2015). Interaction design patterns for spatio-temporal annotations in video learning environments. In Proceedings of the 20th European Conference on Pattern Languages of Programs (S. 16:1-16:21). ACM. doi: 10.1145/2855321.2855338
- Seidel, N. (2015). Making web video accessible – interaction design patterns for assistive video learning environments. In Proceedings of the 20th European Conference on Pattern Languages of Programs (S. 17:1-17:16). New York, NY, USA: ACM. doi: 10.1145/2855321.2855339

- Seidel, N. (2016). Analyse von GUI-Komponenten in Videolernumgebungen. In U. Lucke, A. Schwill, & R. Zender (Hrsg.), DeLFI 2016 – Die 14. e-Learning Fachtagung Informatik der Gesellschaft für Informatik (S. 289-292). Bonn: Gesellschaft für Informatik.
- Seidel, N. (2016). Video Assessment Patterns: Designing Video Player for Integrated Assessment. In Proceedings of the 21st European Conference on Pattern Languages of Programs (S. 32:1–32:20). New York, NY, USA: ACM. doi: 10.1145/3011784.3011820
- Seidel, N. (2017). Empirical Evaluation Methods for Pattern Languages: Sketches, Classification, and Network Analysis. In Proceedings of the 22st European Conference on Pattern Languages of Programs (S. 13:1–13:24). New York, NY, USA: ACM. doi: 10.1145/3147704.3147719

### **Software (Auswahl):**

- VI-TWO: <https://github.com/nise/vi-two>
- VI-LAB: <https://github.com/nise/vi-lab>
- VI-WIKI: <https://github.com/nise/vi-wiki>
- VI-PRESS <https://github.com/nise/vi-press>
- VI-ANALYTICS: <https://github.com/nise/vi-analytics>
- VI-THERESIENSTADT: <https://github.com/nise/vi-theresienstadt>
- Wissensbasis für Patterns und Videolernumgebungen: <https://github.com/nise/designing-video-interfaces>





# Literaturverzeichnis

- C. Alexander. *The Timeless Way of Building*. Oxford University Press, New York, 1979.
- C. Alexander. *The Nature of Order - The Phenomenon of Life*. Center for Environmental Structure, 2002.
- C. Alexander. New concepts in Complexity Theorie arising from studies in the field of architecture. 2003. URL <http://natureoforder.com/library/scientific-introduction.pdf> (abgerufen am 1.06.2016).
- C. Alexander, S. Ishikawa und M. Silverstein. *A Pattern Language*. Oxford University Press, New York, 1977.
- J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11): 832–844, 1983. doi: 10.1145/182.358434.
- E. Alpay und S. Gulati. Student-led podcasting for engineering education. *European Journal of Engineering Education*, 35(4):415–427, 2010. doi: 10.1080/03043797.2010.487557.
- Apple Computers Inc. *Macintosh human interface guidelines*. Addison-Wesley Publishing Company, 1992.
- R. Arnold, C. Gómez und J. Kammerer. *Selbstlernkompetenz. Arbeitspapier 1 des Forschungsprojektes „Selbstlernfähigkeit, pädagogische Professionalität und Lernkulturwandel“*. TU Kaiserslautern, Kaiserslautern, 2000.
- E. Aronson, D. Bridgeman und R. Geffner. Interdependent interactions and prosocial behavior. *Journal of Research and Development in Education*, 12:16–27, 1978.
- M. Arvola. Interaction Design Patterns for Computers in Sociable Use. *International Journal on Computer Application Technology*, 25(2/3):128–139, 2006. doi: 10.1504/IJCAT.2006.009063.
- ATS. VideoAnt. Academic Technology Services. University of Minnesota, 2007. URL <https://ant2.cehd.umn.edu/> (abgerufen am 1.06.2016).
- O. Aubert und Y. Prié. Advene: Active reading through hypervideo. In *HT 2005 16th ACM Conference on Hypertext and Hypermedia*, pages 235–244, 2005. doi: 10.1145/1083356.1083405.
- P. Avgeriou, A. Papasalouros und S. Retalis. Patterns for Designing Learning Management Systems. In K. Henney und D. Schütz, editors, *Proceedings of the 8th European Conference on Pattern Languages of Programs*, pages 115–132. UVK - Universitätsverlag Konstanz, 2003a.
- P. Avgeriou, A. Papasalouros, S. Retalis und M. Skordalakis. Towards a Pattern Language for Learning Management Systems. *Journal of Educational Technology & Society*, 6(2):11–24, 2003b.
- R. Baggetun, E. Rusman und C. Poggi. Design Patterns For Collaborative Learning : From Practice To Theory And Back. In *World Conference on Educational Media and Technology, EDMEDIA'04*, 2004.
- A. W. Bates. *Broadcasting in Education: An Evaluation*. Constable, London, 1984.
- A. W. Bates. *Teaching in a digital age – Guidelines for designing , teaching and learning*. open.bccampus.ca, 2015. URL <http://opentextbc.ca/teachinginadigitalage/> (abgerufen am 1.06.2016).
- R. Bauer und P. Baumgartner. The Potential of Christopher Alexander’s Theory and Practice of Wholeness: Clues for Developing an Educational Taxonomy. In *Proceedings of the 15th European Conference on Pattern Languages of Programs, EuroPLoP '10*, pages 12:1–12:21, New York, NY, USA, 2010. ACM. doi: 10.1145/2328909.2328924.
- R. Bauer und P. Baumgartner. Showcase of Learning: Towards a Pattern Language for Working with Electronic Portfolios in Higher Education. In *Proceedings of the 16th European Conference on Pattern Languages of Programs, EuroPLoP '11*, pages 6:1–6:30, New York, NY, USA, 2012. ACM. doi: 10.1145/2396716.2396722.
- B. Baxley. Universal Model of a User Interface. In *Proceedings of the 2003 Conference on Designing for User Experiences, DUX '03*, pages 1–14, New York, NY, USA, 2003. ACM. doi: 10.1145/997078.997090.

- F. Benevenuto, F. Duarte, T. Rodrigues, V. A. F. Almeida, J. M. Almeida und K. W. Ross. Understanding video interactions in youtube. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, pages 761–764, New York, NY, USA, 2008. ACM. doi: 10.1145/1459359.1459480.
- A. Berger, R. Moretti, P. Chastonay, P. Dillenbourg, A. Bchir, R. Baddoura, C., Bengondo, D. Scherly, P. Ndumbe, P. Farah und B. Kayser. Teaching community health by exploiting international socio-cultural and economical differences. In P. Dillenbourg, A. Eurelings und K. Hakkarainen, editors, *Proceedings of the first European Conference on Computer Supported Collaborative Learning*, pages 97–105, Maastricht, 2001.
- J. Bergin, J. Eckstein, M. L. Manns, H. Sharp, K. Marquardt, J. Chandler, M. Sipos, M. Voelter und E. Wallingford. *Pedagogical Patterns: Advice For Educators*. CreateSpace, 2012.
- J. Bergmann und A. Sams. *Flip your classroom: Reach every student in every class every day*. International Society for Technology in Education, 2012.
- R. Berjon, T. Leithead, E. D. Navara, E. O'Connor und S. Pfeiffer. HTML5 – A vocabulary and associated APIs for HTML and XHTML – W3C Candidate Recommendation 17 December 2012, 2012. URL <http://www.w3.org/TR/2012/CR-html5-20121217/> (abgerufen am 1.06.2016).
- M. Bernstein. Patterns of hypertext. In *Proceedings of ACM Hypertext '98*, pages 21–29, New York, 1998. ACM Press. doi: 10.1145/276627.276630.
- W. Bhikharie, M. van de Watering, H. Huurdeman, W. den Boer, S. Bruins und A. Eliëns. XIMPEL – a web-based framework for interactive video, between storytelling and gameplay, 2016. URL <http://ximpel.net/> (abgerufen am 1.06.2016).
- A. Birukou. A Survey of Existing Approaches for Pattern Search and Selection. In *Proceedings of the 15th European Conference on Pattern Languages of Programs*, EuroPLoP '10, pages 2:1–2:13, New York, NY, USA, 2010. ACM. doi: 10.1145/2328909.2328912.
- J. L. Bishop und M. A. Verleger. The flipped classroom: A survey of the research. In *ASEE National Conference Proceedings, Atlanta, GA*, 2013.
- S. Björk, S. Lundgren, H. Grauers und S. Göteborg. Game Design Patterns. In *in Level Up: Digital Games Research Conference 2003*, pages 4–6, Utrecht, 2003. Faculty of Arts, Utrecht University.
- A. M. Blessing und U. Kortenkamp. VideoClipQuests - Eine flexible Methode für Unterricht und E-Learning. *L.A. multimedia*, Heft 4:8–10, 2009.
- A. M. Blessing und U. Kortenkamp. VideoClipQuests as a New Setup for Learning. In J. Spector, Michael, D. Ifenthaler, P. Isaías, K. Sampson und Demetrios, editors, *Learning and Instruction in the Digital Age*, pages 145–157. Springer US, 2010.
- J. Borchers. *A Pattern Approach to Interaction Design*. Wiley, 2001.
- J. Borchers. Teaching HCI Patterns: Experience from Two University Courses. In *CHI 2002 International Conference on Human Factors and Computing Systems*. ACM, apr 2002.
- J. O. Borchers. A Pattern Approach to Interaction Design. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '00, pages 369–378, New York, NY, USA, 2000. ACM. doi: 10.1145/347642.347795.
- J. Börner, J. Buraczynska, J. Gärtner, K. Nolte, J. Priesmeyer, A. Üstek, C. Stephan, S. Strasdat, S. Manske und H. U. Hoppe. Fake or Real? Analyse physikalischer Phänomene in viralen Videos im forschend-entdeckenden Lernkontext. In U. Lucke, A. Schwill und R. Zender, editors, *Die 14. E-Learning Fachtagung Informatik, Lecture Notes in Informatics (LNI)*, Bonn, 2016. Gesellschaft für Informatik.
- J. Breuer. Lernmethoden, 2000. URL [http://www.forschungsnetzwerk.at/downloadpub/2000\\_breuer\\_Lernmethoden.pdf](http://www.forschungsnetzwerk.at/downloadpub/2000_breuer_Lernmethoden.pdf) (abgerufen am 1.06.2016).
- R. Briggs und G.-J. de Vreede. *ThinkLets: Building Blocks for Concerted Collaboration*. Lulu, Omaha, 2009.

- C. Brooks, C. Thompson und J. Greer. Visualizing Lecture Capture Usage: A Learning Analytics Case Study. In *Workshop on Analytics on Video-Based Learning at 3rd Conference on Learning Analytics and Knowledge 2013*, pages 9–14, Leuven, 2013. ACM.
- M. C. Buchanan und P. T. Zellweger. Automatic temporal layout mechanisms revisited. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1):60–88, feb 2005. doi: 10.1145/1047936.1047942.
- D. Bulterman, J. Jansen, P. Cesar, S. Mullender, E. Hyche, M. DeMeglio, J. Quint, H. Kawamura, D. Weck, X. G. Pañeda, D. Melendi, S. Cruz-Lara, M. Hanclik, D. F. Zucker und T. Michel. Synchronized Multimedia Integration Language (SMIL 3.0) – W3C Recommendation 01 December 2008, 2008. URL <https://www.w3.org/TR/REC-smil/> (abgerufen am 1.06.2016).
- D. Burlak. *Analyse, Design und Implementierung von algorithmenbasierter Lerngruppen-Optimierung*. Masterarbeit, Technische Universität Darmstadt, 2013.
- F. Buschmann und K. Henney. Pattern-Oriented Software Architecture. *OOP 2008 - Software meets business*, 2008.
- V. Bush. As We May Think. *Atlantic Monthly*, 176:101–108, 1945. URL <http://www.theatlantic.com/doc/194507/bush> (abgerufen am 1.06.2016).
- B. Caldwell, M. Cooper, L. G. Reid, G. Vanderheiden, W. Chisholm, J. Slatin und J. White. Web Content Accessibility Guidelines (WCAG) 2.0. Bericht, W3C, 2008. URL <http://www.w3.org/TR/WCAG20/> (abgerufen am 1.06.2016).
- R. A. Calvo und A. Turani. E-Learning Frameworks = (Design Patterns + Software Components). In P. Goodyear und S. Retalis, editors, *Technology-Enhanced Learning - Design Patterns and Pattern Languages*, pages 123–138. Sense Publishers, Rotterdam, 2. Auflage, 2010.
- F. Cazenave, V. Quint und C. Roisin. Timesheets.js: When SMIL Meets HTML5 and CSS3. In *Proceedings of the 11th ACM Symposium on Document Engineering*, pages 43–52, New York, NY, USA, 2011. ACM. doi: 10.1145/2034691.2034700.
- T. Chambel, C. Zahn und M. Finke. Hypervideo design and support for contextualized learning. *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, pages 345–349, 2004. doi: 10.1109/ICALT.2004.1357433.
- T. Chambel und N. Guimarães. Communicating and Learning Mathematics with Hypervideo. In J. Borwein, J., Morales, M.H., Polthier, K., Rodrigues, editor, *Multimedia Tools for Communicating Mathematics*, Mathematics and Visualization, chapter 6, pages 79–92. Springer, 2002.
- A. Chatzigeorgiou, N. Tsantalis und I. Deligiannis. An empirical study on students’ ability to comprehend design patterns. *Computers & Education*, 51(3):1007–1016, 2008. doi: 10.1016/j.compedu.2007.10.003.
- N. Chen und M. Rabb. A pattern language for screencasting. In *Proceedings of the 16th Conference on Pattern Languages of Programs*, PLoP ’09, pages 6:1–6:11, New York, 2010. ACM. doi: 10.1145/1943226.1943234.
- N. Chomsky. *Syntactic Structures*. Walter de Gruyter & Co, Berlin/New York, 2. Auflage, 1957.
- K. Chorianopoulos und M. N. Giannakos. Usability Design for Video Lectures. In *Proceedings of the 11th European Conference on Interactive TV and Video*, EuroITV ’13, pages 163–164, New York, NY, USA, 2013. ACM. doi: 10.1145/2465958.2465982.
- Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2013-2018, 2014. URL [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white\\_paper\\_c11-481360.pdf](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf) (abgerufen am 1.06.2016).
- H. H. Clark und S. E. Brennan. Grounding in Communication. In L. B. Resnick, J. M. Levine und S. D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association, 1991. URL [http://psych.stanford.edu/\\$\sim\\$herb/](http://psych.stanford.edu/$\sim$herb/) (abgerufen am 1.06.2016).
- R. C. Clark, F. Nguyen und J. Sweller. *Efficiency in Learning - Evidence-Based Guidelines to Manage Cognitive Load*. Pfeiffer, San Francisco, 2006.

- J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37, 1960. doi: 10.1177/001316446002000104.
- R. Corfman. An Overview of Patterns. In *The pattern handbook*, pages 19–30. Cambridge University Press, Cambridge, 1998.
- C. Crumlish und E. Malone. *Designing Social Interfaces*. O'Reilly Media, Sebastopol, 2009.
- M. Cutumisu, C. Onuczko, D. Szafron, J. Schaeffer, M. McNaughton, T. Roy, J. Siegel und M. Carbonaro. Evaluating Pattern Catalogs: The Computer Games Experience. In *Proceedings of the 28th International Conference on Software Engineering, ICSE '06*, pages 132–141, New York, NY, USA, 2006. ACM. ISBN 1-59593-375-1. doi: 10.1145/1134285.1134305. URL <http://doi.acm.org/10.1145/1134285.1134305> (abgerufen am 1.06.2016).
- J. L. Cybulski und T. Linden. Composing Multimedia Artefacts for Reuse. In *Proceedings of Pattern Languages of Programs'98*, page 15, 1998. URL [http://hillside.net/plop/plop98/final\\_submissions/P38.pdf](http://hillside.net/plop/plop98/final_submissions/P38.pdf) (abgerufen am 1.06.2016).
- M. Dale, A. Stern, M. Deckert und W. Sack. System demonstration: Metavid.org: a social website and open archive of congressional video. In *Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government*, pages 309–310, Puebla, Mexico, 2009. Digital Government Society of North America.
- N. de Moel und G. van der Veer. Design pattern based decision support. In *Proceedings of ECCE 2011 Conference*, pages 93–96, 2011. doi: 10.1145/2074712.2074732.
- M. Debevc, R. Šafarič und M. Golob. Hypervideo application on an experimental control system as an approach to education. *Computer Applications in Engineering Education*, 16(1):31–44, 2008. doi: 10.1002/cae.20116.
- D. E. DeLano. Patterns Mining. In L. Rising, editor, *The Pattern Handbook*, pages 87–96. Cambridge University Press, Cambridge, 1998.
- M. Derntl und R. Motschnig-Pitrik. A Pattern Approach to Person-Centered e-Learning Based on Theory-Guided Action Research. In *Networked Learning Conference 2004*, page k.A., Lancaste, 2004. URL [http://www.networkedlearningconference.org.uk/past/nlc2004/proceedings/symposia/symposium12/derntl\\_pitrik.htm](http://www.networkedlearningconference.org.uk/past/nlc2004/proceedings/symposia/symposium12/derntl_pitrik.htm) (abgerufen am 1.06.2016).
- S. DeRose, E. Maler und D. Orchard. XML Linking Language (XLink) Version 1.0. Bericht, World Wide Web Consortium, 2001. URL <http://www.w3.org/TR/xlink/> (abgerufen am 1.06.2016).
- Deutsches Institut für Normung (Hsg.). DIN EN ISO 9241, 2003.
- Deutsches Institut für Normung (Hsg.). DIN ISO 14915-110, 2006.
- P. Dillenbourg. What do you mean by 'collaborative learning'? In P. Dillenbourg, editor, *Collaborative-learning: Cognitive and Computational Approaches*, pages 1–15, Oxford, 1999. Elsevier.
- P. Dillenbourg. Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. Kirschner, editor, *Three worlds of CSCL. Can we support CSCL?*, pages 61–91. Open Universiteit Nederland, Heerlen, 2002.
- P. Dillenbourg. Framework for integrated learning. Bericht December, École Polytechnique Fédérale de Lausanne, Lausanne, 2004. URL <http://telearn.archives-ouvertes.fr/docs/00/19/01/07/PDF/Dillenbourg-Kaleidoscope-2004.pdf> (abgerufen am 1.06.2016).
- P. Dillenbourg. ManyScripts, 2006. URL <http://manyscripts.epfl.ch/> (abgerufen am 1.06.2016).
- P. Dillenbourg und F. Hong. The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning*, 3(1):5–23, jan 2008. doi: 10.1007/s11412-007-9033-1.
- P. Dillenbourg und P. Tchounikine. Flexibility in macro-scripts for CSCL. *Journal of Computer Assisted Learning*, 23(1):1–13, 2007. doi: 10.1111/j.1365-2729.2007.00191.x.
- M. Dolan, G. Freed, S. Hayes, E. Hodge, D. Kirby, T. Michel und D. Singer. Timed Text Markup Language (TTML) 1.0 (Second Edition) W3C Proposed Edited Recommendation 09 July 2013, 2013. URL <http://www.w3.org/TR/2013/PER-ttbf1-dfxp-20130709/> (abgerufen am 1.06.2016).



- N. Döring und J. Bortz. *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. Springer-Verlag, Berlin Heidelberg, 5 Auflage, 2006. doi: 10.1007/978-3-642-41089-5.
- A. Duda und C. Keraman. Structured Temporal Composition of Multimedia Data. In *Multi-Media Database Management Systems, 1995. Proceedings., International Workshop on Multi-Media Database Management Systems*, pages 136–148, 1995. doi: 10.1109/MMDBMS.1995.520432.
- J. S. Dumas und J. Redish. *A practical guide to usability testing*. Intellect Books, Exeter, 1999.
- M. Engeli. *Bits and spaces: architecture and computing for physical, virtual, hybrid realms : 33 projects by Architecture and CAAD, ETH Zurich. Volume 1*. Birkhäuser, 2001.
- Enounce. MySpeed. <http://www.enounce.com/myspeed>, 2014. URL <http://www.enounce.com/myspeed> (abgerufen am 1.06.2016).
- B. Ertl. *Kooperatives Lernen in Videokonferenzen*. Dissertation, Ludwig Maximilian Universität München, 2003.
- B. Ertl und H. Mandl. Kooperationsskripts als Lernstrategie. Bericht November, Ludwig-Maximilians-Universität München, Department Psychologie, Institut für Pädagogische Psychologie, München, 2004.
- A. Evseeva und A. Solozhenko. Use of Flipped Classroom Technology in Language Learning. *Procedia - Social and Behavioral Sciences*, 206:205–209, 2015. doi: 10.1016/j.sbspro.2015.10.006.
- S. Fincher. Analysis of Design: An Exploration of Patterns and Pattern Languages for Pedagogy. *Journal of Computers in Mathematics and Science Teaching*, 18(3):331–348, 1999.
- S. Fincher. Perspectives on HCI patterns: concepts and tools (introducing PLML). *Interfaces*, 56: 26–28, 2003.
- M. Finke. *Unterstützung des kooperativen Wissenserwerbs durch Hypervideo-Inhalte*. Dissertation, Technische Universität Darmstadt, 2005.
- J. Finlay, J. Gray, I. Falconer, J. Hensman, Y. Mor und S. Warburton. The Pattern Language Network. Taming web2.0 in Higher Education, 2010. URL <https://patternlanguagenetwork.googlecode.com/files/PatternLanguageNetworkFeb2010-XWiki0ctBackup.pdf> (abgerufen am 1.06.2016).
- P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954. doi: 10.1037/h0055392.
- E. Gamma, R. Helm, R. Johnson und J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- M. Giannakos, L. Jaccheri und J. Krogstie. Looking at MOOCs Rapid Growth Through the Lens of Video-Based Learning Research. *International Journal of Emerging Technologies in Learning (iJET)*, 9 (1), 2014. URL <http://online-journals.org/index.php/i-jet/article/view/3349> (abgerufen am 1.06.2016).
- A. Girgensohn, F. Shipman und L. Wilcox. Hyper-Hitchcock: Authoring Interactive Videos and Generating Interactive Summaries. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 92–93, New York, NY, USA, 2003. ACM Press. doi: 10.1145/957013.957030.
- A. Girgensohn, L. Wilcox, F. Shipman und S. Bly. Designing affordances for the navigation of detail-on-demand hypervideo. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 290–297, New York, NY, USA, 2004. ACM Press. doi: 10.1145/989863.989913.
- P. Glader. Beyond Khan Academy: YouTube Markets Other Education Rockstar Videos. *Wired Academic*, 2012. URL <http://www.wiredacademic.com/2012/02/new-channels-make-education-a-fast-growing-content-category-on-youtube/> (abgerufen am 1.06.2016).
- A. Goeze, J. Zottmann, F. Vogel, F. Fischer und J. Schrader. Getting immersed in teacher and student perspectives? Facilitating analytical competence using video cases in teacher education. *Instructional Science*, 42(1):91–114, 2014. doi: 10.1007/s11251-013-9304-3.

- D. R. Goldman. *A Framework for Video Annotation, Visualization, and Interaction*. University of Washington, Seattle, WA, USA, 2007.
- P. Goodyear, M. de Laat und V. Lally. Using Pattern Languages to Mediate Theory-Praxis Conversations in Design for Networked Learning. *ALT-J: Research in Learning Technology*, 14(3): 211–223, 2006. doi: 10.1080/09687760600836977.
- Google. blockly. <https://developers.google.com/blockly/>, 2015. URL <https://developers.google.com/blockly/> (abgerufen am 1.06.2016).
- S. L. Greene, P. M. Matchen, L. Jones, J. C. Thomas und M. Callery. Tool-based Decision Support for Pattern Assisted Develop. In *Perspectives on HCI Patterns: Concepts and Tools Workshop at CHI 2003*, 2003.
- R. N. Griffiths und L. Pemberton. Teaching Usability Design Through Pattern Language, 1998. URL <http://www.it.bton.ac.uk/staff/lp22/CHIpaper.html> (abgerufen am 1.06.2016).
- R. Grigoras, V. Charvillat und M. Douze. Optimizing Hypervideo Navigation using a Markov Decision Process Approach. In *MULTIMEDIA '02: Proceedings of the eleventh ACM international conference on Multimedia*, pages 39–48, 2002. doi: 10.1145/641007.641014.
- A. Groß, C. Meinel, F. Moritz und M. Siebert. Aufbau eines Multi-Plattform-Lernvideo-Archivs: Herausforderungen und Lösungen. *Handbuch E-Learning*, 31:3747–3766, 2010.
- T. Gross. Mensch-Computer-Interaktion. In J. Haake, G. Schwabe und M. Wessner, editors, *CSCL-Kompodium 2.0*, chapter 1.8, pages 77–82. Oldenbourg Wissenschaftsverlag Verlag, München, 2012.
- F. Halasz und M. Schwartz. The Dexter hypertext reference model. *Commun. ACM*, 37(2):30–39, feb 1994. doi: 10.1145/175235.175237.
- L. Hardman. *Modelling and Authoring Hypermedia Documents*. Phd thesis, University of Amsterdam, 2004. URL [http://homepages.cwi.nl/~sim\\$lynda/thesis/Chap3.pdf](http://homepages.cwi.nl/~sim$lynda/thesis/Chap3.pdf) (abgerufen am 1.06.2016).
- L. Hardman, J. Van Ossenbruggen, K. Mullender, L. Rutledge und D. Bulterman. Do you have the time? Composition and linking in time-based hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots*, pages 189–196. ACM, 1999. doi: 10.1145/294469.294515.
- A. Harrer und N. Malzahn. Bridging the Gap - Towards a Graphical Modeling Language for Learning Designs and Collaboration Scripts of Various Granularities. In *Advanced Learning Technologies, 2006. Sixth International Conference on Advanced Learning Technologies*, pages 296–300, Washington, DC, USA, 2006. IEEE Computer Society. doi: 10.1109/ICALT.2006.1652428.
- A. Harrer, L. Kobbe und N. Mazahn. Conceptual and Computational Issues in the Formalization of Collaboration Scripts. *Computer-Supported Collaborative Learning*, pages 280–282, 2007.
- A. Harrer, D. Kohen-vacs, B. Roth und N. Malzahn. Design and Enactment of Collaboration Scripts – an integrative approach with graphical notations and learning platforms Background – Computer-enacted collaboration scripts. In A. Dimitracopoulou, C. O'Malley, D. Suthers und P. Reimann, editors, *Computer supported collaborative learning practices : CSCL2009 community events proceedings*, pages 198–200, Rhodos, 2009. International Society of the Learning Sciences (ISLS).
- N. Harrison. The Language of Shepherds: A Pattern Language for Shepherding. In *Proceedings of the 6th Annual Conference on the Pattern Languages of Programs (PLoP)*, Monticello, Illinois, USA., aug 1999. URL [http://jerry.cs.uiuc.edu/~sim\\$plop/plop99/proceedings/harrison/shepherding4.pdf](http://jerry.cs.uiuc.edu/~sim$plop/plop99/proceedings/harrison/shepherding4.pdf) (abgerufen am 1.06.2016).
- B. Hasler, B. Kersten und J. Sweller. Learner control, cognitive load and instructional animation. *Applied Cognitive Psychology*, 21(6):713–729, 2007. doi: 10.1002/acp.1345.
- G. Hatsidimitris und B. Allen. An Online Resource for the Design of Instructional Videos and Animations. In J. Sanchez und K. Zhang, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2010*, pages 1024–1028, Orlando, Florida, USA, oct 2010. AACE. URL <http://www.editlib.org/p/35684> (abgerufen am 1.06.2016).

- M. Hausenblas, P. Jägenstedt, J. Jansen, Y. Lafon, C. Parker und T. Steiner. Media Fragments URI 1.0 – W3C Working Draft 30 November 2011, 2011. URL <http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-spec/> (abgerufen am 1.06.2016).
- S. Henninger, K. Haynes und M. W. Reith. A Framework for Developing Experience-based Usability Guidelines. In *Proceedings of the 1st Conference on Designing Interactive Systems: Processes, Practices, Methods, & Techniques*, DIS '95, pages 43–53, New York, NY, USA, 1995. ACM. doi: 10.1145/225434.225440.
- D. Hernández-Leo, E. D. Villasclaras-Fernández, J. I. Asensio-Pérez und Y. Dimitriadis. COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology & Society*, 9(1):58–71, 2006.
- D. Hernández-Leo. Collage 2, 2012. URL <http://pandora.tel.uva.es/wic2/> (abgerufen am 1.06.2016).
- D. Hernandez-Leo, E. D. Villasclaras-Fernandez, J. I. Asensio-Perez, Y. A. Dimitriadis und S. Retalis. CACL Scripting Patterns: Hierarchical Relationships and Applicability. *Advanced Learning Technologies, IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, 0: 388–392, 2006. doi: <http://doi.ieeeecomputersociety.org/10.1109/ICALT.2006.126>.
- HHS. Research-Based Web Design & Usability Guidelines, 2015. URL <http://guidelines.usability.gov/> (abgerufen am 1.06.2016).
- C. E. Hmelo-Silver, R. Jordan, L. Liu und E. Chernobilsky. Representational Tools for Understanding Complex Computer-Supported Collaborative Learning Environments. In S. Puntambekar, G. Erkens und C. Hmelo-Silver, editors, *Analyzing Interactions in CACL Methods, Approaches and Issues*, chapter 4, pages 83–106. Springer, 2011. doi: 10.1007/978-1-4419-7710-6.
- P. Hoffmann und M. Herczeg. Hypervideo vs. Storytelling Integrating Narrative Intelligence into Hypervideo. In S. Göbel, R. Malkewitz und I. Iurgel, editors, *Technologies for Interactive Digital Storytelling and Entertainment, Third International Conference, TIDSE 2006, Darmstadt, Germany, December 4-6, 2006*, volume 4326 of *Lecture Notes in Computer Science*, pages 37–48, Berlin, 2006. Springer. doi: 10.1007/11944577\_4.
- C. Hofmann und N. Hollender. Kooperativer Informationsaustausch mit Hypervideo: Potentiale für das Web 2.0. In C. Rensing und G. Rößling, editors, *Proceedings der Pre-Conference Workshops der 5. e-Learning Fachtagung Informatik DeLFI 2007*, pages 51–58, Berlin, 2007. Logos Verlag.
- C. Hofmann, N. Hollender und D. W. Fellner. Task- and Process-related Design of Video Annotation Systems. In H. Wandke, S. Kain und D. Struve, editors, *Mensch & Computer 2009: Grenzenlos frei!?*, pages 173–182, München, 2009. Oldenbourg Verlag.
- G. Holden, N. Schadewitz und C. Rapanta. Patterns for the Creation of Elearning Content and Activities in a University Setting. In *Proceedings of the 15th European Conference on Pattern Languages of Programs*, EuroPLoP '10, pages 10:1–10:13, New York, NY, USA, 2010. ACM. doi: 10.1145/2328909.2328922.
- H. J. Holz, A. Applin, B. Haberman, D. Joyce, H. Purchase und C. Reed. Research Methods in Computing: What Are They, and How Should We Teach Them? *ACM SIGCSE Bulletin*, 38(4):96–114, 2006. doi: 10.1145/1189136.1189180.
- T. Hoppe-Seyler, M. Gartmeier, G. Möller, J. Bauer, A. Wiesbeck und G. Karsten. Entwicklung von Lehrfilmen zur Gesprächs- führung zwischen Realitätsnähe und systematischer didaktischer Gestaltung Zusammenfassung. *Zeitschrift für Hochschulentwicklung - Videos in der (Hochschul-)Lehre*, 9(3), 2014. URL <http://www.zfhe.at/index.php/zfhe/article/view/681> (abgerufen am 1.06.2016).
- K. M. Höver, S. Borgert und M. Mühlhäuser. S-BPM ONE - Running Processes: 5th International Conference, S-BPM ONE 2013, Deggendorf, Germany, March 11-12, 2013. *Proceedings*, chapter A Domain S, pages 72–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-36754-0\_5.

- K. Höver und M. Mühlhäuser. Can We Use S-BPM for Modeling Collaboration Scripts? In C. Zehbold, editor, *S-BPM ONE - Application Studies and Work in Progress*, volume 422 of *Communications in Computer and Information Science*, pages 174–187. Springer International Publishing, 2014. doi: 10.1007/978-3-319-06191-7\_13.
- C. Hübscher, S. L. Pauwels, S. P. Roth, J. A. Bargas-Avila und K. Opwis. The Organization of Interaction Design Pattern Languages Alongside the Design Process. *Interact. Comput.*, 23(3): 189–201, 2011. doi: 10.1016/j.intcom.2011.02.009.
- M. Hündgen und B. Gugel. Youtube - Zwischen Goldrausch und Wildem Westen, 2013. URL <http://youtu.be/J7KSmReUU1E> (abgerufen am 1.06.2016).
- H. D. Hussey, B. K. B. Fleck und A. S. Richmond. Promoting active learning through a flipped course design. *Promoting Active Learning through the Flipped Classroom Model*, page 23, 2014. doi: 10.4018/978-1-4666-4987-3.ch002.
- H. Huurdeman, M. van de Watering, W. Bhikharie und A. Eliëns. Introducing XIMPEL - The eXtensible Interactive Media Player for Entertainment and Learning. Bericht, Intelligent Multimedia Group, VU University, Amsterdam, 2008. URL [http://www.timelessfuture.com/archive/documents/Introducing\\_XIMPEL\\_DIVERSE\\_2008.pdf](http://www.timelessfuture.com/archive/documents/Introducing_XIMPEL_DIVERSE_2008.pdf) (abgerufen am 1.06.2016).
- T. Iba. *Pattern Illustrating Patterns - A Pattern Language for Pattern Illustrating*. CreativeShift Lab, Inc., 2015.
- Infragistics. Quince UX Paterns. <http://quince.infragistics.com/html/>, 2013.
- A. Jeffery und S. Currier. What Is... IMS Learning Design?, 2005. URL [http://publications.cetis.org.uk/wp-content/uploads/2011/10/WhatIsLD2%5C\\_web.pdf](http://publications.cetis.org.uk/wp-content/uploads/2011/10/WhatIsLD2%5C_web.pdf) (abgerufen am 1.06.2016).
- P. Jermann und P. Dillenbourg. An analysis of learner arguments in a collective learning environment. In *Proceedings of the Third International Conference on Computer Support for Collaborative Learning*, pages 265–273, 1999. URL <https://hal.archives-ouvertes.fr/hal-00190257/document> (abgerufen am 1.06.2016).
- C. Jödicke und E. Teich. Konzepte für den Einsatz von E-Tutoren in komplexen E-Learning-Szenarien - Ein Erfahrungsbericht. In T. Köhler, N. Kahnwald und E. Schoop, editors, *Wissensgemeinschaften in Wirtschaft und Wissenschaft - Konferenzbeiträge der 18. GeNeMe*, pages 45–53, 2015.
- D. W. Johnson, R. T. Johnson und K. A. Smith. *Active Learning: Cooperation in the College Classroom*. Interaction Book Company, Edina, MN, 3. Auflage, 2006.
- Kaltura-Corporation. Kaltura, 2016. URL <http://corp.kaltura.com/> (abgerufen am 1.06.2016).
- P.-T. Kandzia, S. Linckels, T. Ottmann und S. Trahasch. Lecture Recording - vom Experiment zum Hochschulalltag. In *Deutsche eLearning Fachtagung Informatik*, Bonn, 2011. Gesellschaft für Informatik.
- S. Kane, J. Wobbrock und R. Ladner. Usable gestures for blind people: understanding preference and performance. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, CHI '11, pages 413–422, New York, NY, USA, 2011. ACM. ISBN 9781450302678. doi: 10.1145/1978942.1979001. URL <http://dl.acm.org/citation.cfm?id=1979001> (abgerufen am 1.06.2016).
- K. Kappel, M. Tomitsch, T. Költringer und T. Grechenig. Developing User Interface Guidelines for DVD Menus. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 177–182, New York, NY, USA, 2006. ACM. doi: 10.1145/1125451.1125490.
- R. H. Kay. Exploring the use of video podcasts in education: A comprehensive review of the literature. *Computers in Human Behavior*, 28(3):820–831, 2012. doi: 10.1016/j.chb.2012.01.011.
- M. Kerres. *Mediendidaktik: Konzeption und Entwicklung mediengestützter Lernangebote*. Oldenbourg Wissenschaftsverlag, 3. Auflage, 2012.



- M. Ketterl, R. Mertens und O. Vornberger. Vorlesungsaufzeichnungen 2.0. In *Lernen - Organisation - Gesellschaft. eCampus-Symposium der Osnabrücker Hochschulen*, pages 2–5, Osnabrück, 2008. EPOS-Verlag.
- M. Ketterl, O. A. Schulte und A. Hochman. Opencast Matterhorn: A Community-Driven Open Source Solution for Creation, Management and Distribution of Audio and Video in Academia. *ISM '09: Proceedings of the 2009 11th IEEE International Symposium on Multimedia*, 11:687–692, 2009.
- C. D. Khazaeli. *Systemisches Design*. Rowohlt, Reinbeck bei Hamburg, 2005.
- J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos und R. C. Miller. Understanding In-video Dropouts and Interaction Peaks Inonline Lecture Videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference, L@S '14*, pages 31–40, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2669-8. doi: 10.1145/2556325.2566237. URL <http://doi.acm.org/10.1145/2556325.2566237> (abgerufen am 1.06.2016).
- L. Kobbe, A. Weinberger, P. Dillenbourg, A. Harrer, R. Härmäläinen, P. Häkkinen und F. Fischer. Specifying computer-supported collaboration scripts. *International Journal of Computer-Supported Collaborative Learning*, 2(2-3):211–224, sep 2007. doi: 10.1007/s11412-007-9014-4.
- C. Kohls. *The Theories of Design Patterns and their Practical Implications exemplified for E-Learning Patterns*. Dissertation, Katholische Universität Eichstätt-Ingolstadt, 2014.
- C. Kohls und S. Panke. Is That True...?: Thoughts on the Epistemology of Patterns. In *Proceedings of the 16th Conference on Pattern Languages of Programs, PLoP '09*, pages 9:1–9:14, New York, NY, USA, 2010. ACM. doi: 10.1145/1943226.1943237.
- C. Kohls und J.-G. Uttecht. Lessons learnt in mining and writing design patterns for educational interactive graphics. *Computers in Human Behavior*, 25(5):1040–1055, 2009. doi: 10.1016/j.chb.2009.01.004.
- S. Köhne. *Didaktischer Ansatz für das Blended Learning : Konzeption und Anwendung von Educational Patterns*. Dissertation, Hohenheim, 2005.
- I. Kollar, F. Fischer und F. Hesse. Collaboration Scripts – A Conceptual Analysis. *Educational Psychology Review*, 18(2):159–185, 2006. ISSN 1040-726X. doi: 10.1007/s10648-006-9007-2.
- T. Költringer, M. Tomitsch, K. Kappel und T. Grechenig. Guidelines for Designing Usable DVD Menus. In *The 20th BCS HCI Group conference*, 2005.
- J. Konert, D. Burlak, S. Göbel und R. Steinmetz. GroupAL: ein Algorithmus zur Formation und Qualitätsbewertung von Lergruppen in E-Learning-Szenarien mittels n-dimensionaler Gütekriterien. In A. Breiter und C. Rensing, editors, *DeLFI 2013 – Die 11. e-Learning Fachtagung Informatik*, pages 71–82, Bonn, 2013. Gesellschaft für Informatik (GI e.V.).
- C. Köppe. Continuous Activity: A Pedagogical Pattern for Active Learning. In *Proceedings of the 16th European Conference on Pattern Languages of Programs, EuroPLoP '11*, pages 3:1–3:7, New York, NY, USA, 2012a. ACM. doi: 10.1145/2396716.2396719.
- C. Köppe. A Pattern Language for Teaching Design Patterns (Part 1). In *Proceedings of the 16th European Conference on Pattern Languages of Programs, EuroPLoP '11*, pages 2:1–2:21, New York, NY, USA, 2012b. ACM. doi: 10.1145/2396716.2396718.
- U. Kortenkamp und A. M. Blessing. VideoClipQuests as an E-Learning Pattern. In C. Kohls und J. Wedekind, editors, *Investigations of E-Learning Patterns: Context Factors, Problems and Solutions*, pages 237–246. IGI Global, 2011.
- K. Koukouletsos, B. Khazaei, A. Dearden und M. Ozcan. Teaching Usability Principles with Patterns and Guidelines. In P. Kotze, W. Wong, J. Jorge, A. Dix und P. Silva, editors, *Creativity and HCI: From Experience to Design in Education*, volume 289 of *IFIP - International Federation for Information Processing*, pages 159–174. Springer US, 2009. doi: 10.1007/978-0-387-89022-7\_11.
- J. Koumi. *Designing Video and Multimedia for open and flexible learning*. Routledge, Abingdon, 2006.
- H. Kromrey. *Empirische Sozialforschung - Messung und Datengewinnung*. Fernuniversität Hagen, 2011.

- M. Krueger. *Das Lernszenario VideoLern: Selbstgesteuertes und kooperatives Lernen mit Vorlesungsaufzeichnungen*. Dissertation, Universität der Bundeswehr München, 2010.
- C. Kruschitz und M. Hitz. Are Human-computer Interaction Design Patterns Really Used? In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 711–714, New York, NY, USA, 2010. ACM. doi: 10.1145/1868914.1869011.
- T. Kunert. *User-Centered Interaction Design Patterns for Interactive Digital Television Applications*. Springer, Dordrecht / Heidelberg / London / New York, 2009.
- J. Kürsten. Metadatenstandards und -formate für audiovisuelle Inhalte. In M. Eibl, J. Kürsten und M. Ritter, editors, *Workshop Audiovisuelle Medien*, pages 182–191. TU Chemnitz, 2009.
- F. Laker. gui.de, 2013. URL <http://www.gui.de> (abgerufen am 1.06.2016).
- J. Lammi, M. Varjokallio und J. Hocksell. patternry.com, 2014. URL <http://patternry.com> (abgerufen am 1.06.2016).
- J. R. Landis und G. G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), 1977.
- T. Lauer und S. Trahasch. Strukturierte verankerte Diskussion als Form kooperativen Lernens mit eLectures. *i-com Zeitschrift für interaktive und kooperative Medien*, 4(3):48–51, 2005a. doi: 10.1524/icom.2005.4.3.48.
- T. Lauer und S. Trahasch. Strukturierte verankerte Diskussion als Form kooperativen Lernens mit eLectures. In U. Lucke, K. Nölting und D. Tavangarian, editors, *Workshop Proceedings DeLFI 2005 und GMW05*, pages 37–42, Berlin, 2005b. Logos Verlag.
- F. Lehner und B. Siegel. E-Learning mit interaktiven Videos - Prototypisches Autorensystem und Bewertung von Anwendungsszenarien. In A. Schwill und N. Apostolopoulos, editors, *Lernen im digitalen Zeitalter – DeLFI 2009 – die 7. E-Learning-Fachtagung Informatik*, pages 43–54, Bonn, 2009. Gesellschaft für Informatik.
- M. Leidel, S. Niemann, H. A. Saliha, J. Cullmann, N. Seidel, D. Borchardt, C. Bernhofer und P. Krebs. International viewpoint and news – IWRM education: e-learning module on Integrated Water Resources Management. *Environmental Earth Sciences*, 68(2):609–613, 2012. doi: 10.1007/s12665-012-2059-3.
- A. Leijen, I. Lam, L. Wildschut, P. R.-J. Simons und W. Admiraal. Streaming video to enhance students' reflection in dance education. *Computers & Education*, 52(1):169–176, 2009. doi: 10.1016/j.compedu.2008.07.010.
- H. Leitner. *Pattern Theory: Introduction and Perspectives on the Tracks of Christopher Alexander*. CreateSpace, 2015.
- G. Liestøl. Aesthetic and rhetorical aspects of linking video in hypermedia. In *Proceedings of the 1994 ACM European conference on Hypermedia technology - ECHT '94*, number 9, pages 217–223, 1994. doi: 10.1145/192757.286994.
- S. Lindley, R. Corish, E. Kosmack Vaara, P. Ferreira und V. Simbelis. Changing perspectives of time in HCI. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 3211–3214, New York, NY, USA, 2013. ACM. doi: 10.1145/2468356.2479649.
- J. Lonchamp. Supporting synchronous collaborative learning: A generic, multi-dimensional model. *International Journal of Computer-Supported Collaborative Learning*, 1(2):247–276, 2006. doi: 10.1007/s11412-006-8996-7.
- R. Lopes und L. Carriico. Patterns for Time-based Hypermedia Artifacts. In U. Zdun und L. B. Hvatum, editors, *Proceedings of the 11th European Conference on Pattern Languages of Programs*, pages 247–276, Konstanz, 2007. Universitätsverlag Konstanz.
- L. Lu. Analysis of Computer Multi-angle Application and Its Technology Development. In *Proceedings of the Second International Conference on Innovative Computing and Cloud Computing*, ICCCC '13, pages 225:225–225:227, New York, NY, USA, 2013. ACM. doi: 10.1145/2556871.2556920.

- U. Lucke und D. Tavangarian. Lecture as a Service: Dienstbasierte Distribution von Vorlesungsaufzeichnungen. In A. Schwill und N. Apostolopoulos, editors, *Lernen im Digitalen Zeitalter Workshop-Band Dokumentation der Pre-Conference zur DeLFI2009 – Die 7. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.*, pages 181–188, Berlin, 2009. Logos Verlag.
- B. Mandl und H. Ertl. Kooperationsskripts als Lernstrategie. (Forschungsbericht Nr. 172). Bericht, Ludwig-Maximilians-Universität, Department Psychologie, Institut für Pädagogische Psychologie, München, 2004.
- E. Mannens, D. Deursen, R. Troncy, S. Pfeiffer, C. Parker, Y. Lafon, J. Jansen, M. Hausenblas und R. Walle. A URI-based approach for addressing fragments of media resources on the Web. *Multimedia Tools and Applications*, 59(2):691–715, 2012. doi: 10.1007/s11042-010-0683-z.
- L. Manovich. *Language of New Media*. The MIT Press, 2001. URL <http://www.manovich.net/LNM/> (abgerufen am 1.06.2016).
- D. Martin, T. Rodden, M. Rouncefield, I. Sommerville und S. Vill. Finding Patterns in the Fieldwork. In W. Prinz, M. Jarke, Y. Rogers, K. Schmidt und V. Wulf, editors, *European Conference on Computer Supported Cooperative Work*, pages 39–58, Bonn, 2001. Kluwer Academic Press.
- T. Martin, A. Boucher und J.-M. Ogier. Multimodal analysis of recorded video for e-learning. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 1043–1044, New York, NY, USA, 2005. ACM. doi: 10.1145/1101149.1101367.
- R. E. Mayer. *Multimedia Learning - second edition*. Cambridge University Press, New York, 2009.
- R. E. Mayer und R. Moreno. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, 38(1), 2003. doi: 10.1207/S15326985EP3801\_6.
- R. E. Mayer, A. Mathias und K. Wetzell. Fostering Understanding of Multimedia Messages Through Pre-training: Evidence for a Two-Stage Theory of Mental Model Construction. *Journal of Experimental Psychology: Applied*, 8(3):147–154, sep 2002. doi: 10.1037/1076-898X.8.3.147.
- MediaWiki.org. Markup spec, 2015. URL [https://www.mediawiki.org/wiki/Markup\\_spec](https://www.mediawiki.org/wiki/Markup_spec) (abgerufen am 1.06.2016).
- N. Mehta, J. Sicking, E. Graff, A. Popescu, J. Orlow und J. Bell. Indexed Database API - W3C Candidate Recommendation 04 July 2013, 2013.
- B. Meixner. *Annotated Interactive Non-linear Video – Software Suite, Download and Cache Management*. Dissertation, Universität Passau, 2014. URL [https://opus4.kobv.de/opus4-uni-passau/files/222/Meixner\\_Britta.pdf](https://opus4.kobv.de/opus4-uni-passau/files/222/Meixner_Britta.pdf) (abgerufen am 1.06.2016).
- B. Meixner und J. Hoffmann. Intelligent download and cache management for interactive non-linear video. *Multimedia Tools and Applications*, pages 1–44, 2012. doi: 10.1007/s11042-012-1158-1.
- B. Meixner und H. Kosch. Interactive non-linear video: definition and XML structure. In *Proceedings of the 2012 ACM symposium on Document engineering*, DocEng '12, pages 49–58, New York, NY, USA, 2012. ACM. doi: 10.1145/2361354.2361367.
- B. Meixner, B. Siegel, G. Hölbling, F. Lehner und H. Kosch. SIVA Suite - Authoring System and Player for Interactive Non-linear Videos. In *MM'10*, pages 1563–1566, 2010.
- B. Meixner, K. Kandlbinder und B. Siegel. Guidelines für interaktive Videos. Bericht, Lehrstuhl für Informatik mit Schwerpunkt Verteilte Informationssysteme. Universität Passau, 2011. URL [http://siva.uni-passau.de/sites/default/files/Guidelines\\_interaktive\\_Videos.pdf](http://siva.uni-passau.de/sites/default/files/Guidelines_interaktive_Videos.pdf) (abgerufen am 1.06.2016).
- B. Meixner, K. Matusik, C. Grill und H. Kosch. Towards an easy to use authoring tool for interactive non-linear video. *Multimedia Tools and Applications*, pages 1–26, 2012. doi: 10.1007/s11042-012-1218-6.
- M. Merkt, S. Weigand, A. Heier und S. Schwan. Learning with videos vs. learning with print: The role of interactive features. *Learning and Instruction*, 21(6):687–704, 2011. doi: 10.1016/j.learninstruc.2011.03.004.

- R. Mertens. *Hypermediale Navigation in Vorlesungsaufzeichnungen – Nutzung und automatische Produktion hypermedial navigierbarer Aufzeichnungen von Lehrveranstaltungen*. Dissertation, Universität Osnabrück, 2007.
- R. Meyer und M. Pedrotti. Interdisziplinäre Lernkontexte durch annotierte Vorlesungsaufzeichnungen. Potential nutzergenerierten Contents im Bereich der Hochschulbildung. In N. Nistor und S. Schirlitz, editors, *Digitale Medien und Interdisziplinarität*, pages 80–90, Münster, 2015. Waxmann.
- Y. Miao, K. Hoeksema, H. U. Hoppe und A. Harrer. CSCL Scripts : Modelling Features and Potential Use. In *CSCL '05 Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, pages 423 – 432. International Society of the Learning Sciences, 2005. URL <http://dl.acm.org/citation.cfm?id=1149293.1149349> (abgerufen am 1.06.2016).
- G. A. Miller. The Magical Number Seven, Plus or Minus Two Some Limits on Our Capacity for Processing Information. *Psychological Review*, 101(2):343–352, 1956.
- B. J. Millis und P. G. Cottell. *Cooperative Learning for Higher Education Faculty. Series on Higher Education*. ERIC, 1997.
- Y. Mor und S. Warburton. Patterns for using video in MOOCs. In *Proceedings of the 21st European Conference on Pattern Languages of Programs*, page [im Druck], New York, NY, USA, 2016. ACM.
- K. Morisse und T. Sempf. An Approach for Collaboration and Annotation in Video Post-production. In V. Sunderam, G. van Albada, P. Sloot und J. Dongarra, editors, *Computational Science - ICCS 2005*, volume 3516 of *Lecture Notes in Computer Science*, pages 95–112. Springer Berlin / Heidelberg, 2005. doi: 10.1007/11428862\_52.
- Mozilla-Foundation. Mozilla Popcorn Maker [Software]. <http://mozillapopcorn.org/>, 2012. URL <http://mozillapopcorn.org/> (abgerufen am 1.06.2016).
- O. Muller, B. Haberman und D. Ginat. Pattern-Oriented Instruction and its Influence on Problem Decomposition and Solution Construction. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 151–155, Dundee, 2007. ACM. doi: 10.1145/1268784.1268830.
- H. M. Niegemann, S. Hessel, M. Deimann, D. Hochscheid-Mauel, K. Aslanski und G. Kruezberger. *Kompendium E-Learning*. Springer, Heidelberg, 2004.
- H. Niegemann, S. Domagk, S. Hessel, A. Hein, M. Hupfer und A. Zobel. Bewegtbilder: Animation, Video und Simulation. In *Kompendium multimediales Lernen*, X.media.press, chapter 18, pages 239–273. Springer Berlin Heidelberg, Dordrecht, 2008.
- J. Nielsen und R. Molich. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256, New York, NY, USA, 1990. ACM. doi: 10.1145/97243.97281.
- D. A. Norman. *The design of everyday things*. Doubleday Business, New York, 2002.
- D. A. Norman und S. W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.
- A. M. O'Donnel und D. F. Dansereau. Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In R. Miller und N. Hertz-Lazarowitz, editors, *Interactions in cooperative groups. The theoretical anatomy of group learning*, pages 120–141. Cambridge University Press, Cambridge, 1992.
- A. O'Donnell und D. F. Dansereau. Learning from Lectures: Effects of Cooperative Review. *The Journal of Experimental Education*, 61(2):116–125, 1993. doi: 10.1080/00220973.1993.9943856.
- R. Of, T. Document, A. R. E. Invited, T. O. Submit, W. T. Comments, C. Of, A. N. Y. Relevant, P. Rights, W. They, A. R. E. Aware, T. O. Provide, I. N. Addition, T. O. Their, E. As, B. Acceptable, F. O. R. Industrial, U. Purposes, D. International, S. May, O. Have, T. O. Be, C. In, L. Of, T. Potential, T. O. Become, D. To, W. Reference, M. A. Y. Be und M. In. ISO/FDIS 8601: Data elements and interchange formats – Information interchange – Representation of dates and times, 2000.



- N. Özdenir und Y. Güngör. Effects of video podcast technology on peer learning and project quality. *Procedia - Social and Behavioral Sciences*, 2(2):2217–2221, 2010. doi: 10.1016/j.sbspro.2010.03.311.
- A. Paivio. *Mental Representations - A dual coding approach*. Oxford University Press, 1990.
- A. S. Palincsar und A. L. Brown. Reciprocal Teaching of Comprehension-Fostering and Comprehension-Monitoring Activities. *Cognition and Instruction* 1, 1(2):117–175, 1984.
- N. Pattanasri, A. Jatowt und K. Tanaka. Enhancing Comprehension of Events in Video through Explanation-on-Demand Hypervideo. In *Proceedings of the 13th International Multimedia Modeling Conference*, pages 535–544. Springer-Verlag, 2007. doi: 10.1007/978-3-540-69423-6\_52.
- R. Pea. DIVER, 2004. URL <http://diver.stanford.edu> (abgerufen am 1.06.2016).
- R. Pea und R. Lindgren. Video Collaboratories for Research and Education: An Analysis of Collaboration Design Patterns. *IEEE Transactions on Learning Technologies*, 1(4):1–13, 2008. doi: 10.1109/TLT.2009.5.
- R. D. Pea. Video-as-Data and Digital Video Manipulation Techniques for Transforming Learning Sciences Research, Education, and Other Cultural Practices. In J. Weiss, J. Nolan, J. Hunsinger und P. Trifonas, editors, *The International Handbook of Virtual Learning Environments*, pages 1321–1393. Springer Netherlands, 2006. doi: 10.1007/978-1-4020-3803-7\_55.
- S. Pearson und Y. Shen. Context-aware Privacy Design Pattern Selection. In *Proceedings of the 7th International Conference on Trust, Privacy and Security in Digital Business*, TrustBus'10, pages 69–80, Berlin, Heidelberg, 2010. Springer-Verlag.
- K. A. Peppler und Y. B. Kafai. From SuperGoo to Scratch: Exploring Media Creative Production in an Informal Learning Environment. *Journal on Learning, Media, and Technology*, 32(2):149–166, 2007. doi: 10.1080/17439880701343337.
- M. Persike und J.-D. Friedrich. Lernen mit digitalen Medien aus Studierendenperspektive. Sonderauswertung aus dem CHE Hochschulranking für die deutschen Hochschulen. Bericht, Geschäftsstelle Hochschulforum Digitalisierung, Berlin, 2016. URL [http://www.stifterverband.de/pdf/lernen\\_mit\\_digitalen\\_medien\\_aus\\_studierendenperspektive.pdf](http://www.stifterverband.de/pdf/lernen_mit_digitalen_medien_aus_studierendenperspektive.pdf) (abgerufen am 1.06.2016).
- D. Pratt und N. Winters. Learning patterns for the design and deployment of mathematical games, 2007. URL <http://lp.noe-kaleidoscope.org/outcomes/patterns/> (abgerufen am 1.06.2016).
- K. Quibeldey-Cirkel. *Entwurfsmuster: Design Patterns in der objektorientierten Softwaretechnik*. Springer-Verlag, Bonn, Heidelberg, 1999. doi: 10.1007/978-3-642-58527-2.
- M. Rauschenberger, M. P. Cota und J. Thomaschewski. Efficient Measurement of the User Experience of Interactive Products. *International Journal of Artificial Intelligence and Interactive Multimedia*, 2(1):39–45, 2013. doi: 10.9781/ijimai.2013.215.
- W. Reinhardt, D. Engbring und J. Magenheimer. Video-Annotationen in der universitären Lehre. Anwendungsszenarien und koaktive Software. *i-com*, 10(1):58–66, may 2011. doi: 10.1524/icom.2011.0009.
- G. Reinmann. *Didaktisches Design*. Universität der Bundeswehr, München, 2010. URL <http://lernen-unibw.de/studientexte> (abgerufen am 1.06.2016).
- M. Reiserer, B. Ertl und H. Mandl. Fostering collaborative knowledge construction in desktop video-conferencing: effects of content schemes and cooperation scripts in peer teaching settings. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, CSCL '02, pages 379–388. International Society of the Learning Sciences, 2002. URL <http://dl.acm.org/citation.cfm?id=1658616.1658670> (abgerufen am 1.06.2016).
- S. Retalis, P. Georgia und Y. Dimitriadis. Eliciting Design Patterns for E-Learning Systems. *Computer Science Education*, 16(2):105–118, 2006. doi: 10.1080/08993400600773323.
- J. Reynolds. Video Jigsaw. *TSOL Journal*, 10(4), 2001. doi: 10.1002/j.1949-3533.2001.tb00047.x.
- R. Riempp. *Intensionales Beobachtungslernen von Bewegungs- und handlungsabläufen mit interaktivem Video*. Dissertation, Universität Tübingen, 2000.

- L. Rising. Pattern Mining. In S. Zamir, editor, *Handbook of Object Technology*, chapter 38. CRC Press, 1999.
- K. Rockenbach und J. Cämmerer. Abschlussbericht: Einführung eines standortübergreifenden e-learning tools zum Erlernen kommunikativer Kompetenzen in der Medizin. Bericht, Universität Leipzig, Leipzig, 2014.
- M. Rohs und R. Streule. Untersuchungen zum Einsatz von eLectures an Hochschulen – Sichtung eines Forschungsfeldes. In A. Schwill und N. Apostolopoulos, editors, *Lernen im Digitalen Zeitalter Workshop-Band Dokumentation der Pre-Conference zur DeLFI2009 – Die 7. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.*, pages 189–196, Berlin, 2009. Logos Verlag.
- A. Rüping. *Where Code and Content Meet. Design Patterns for Web Content Management and Delivery, Personalisation and User Participation*. John Wiley & Sons, Chichester, 2009.
- H. Sack und J. Waitelonis. Zeitbezogene kollaborative Annotation zur Verbesserung der inhaltsbasierten Videosuche. In T. H. S. P. B. Gaiser, editor, *Good Tags – Bad Tags. Social Tagging in der Wissensorganisation*, pages 107–117, Münster, 2008. Waxmann.
- M. Sadallah, O. Aubert und Y. Prié. CHM: an annotation- and component-based hypervideo model for the Web. *Multimedia Tools and Applications*, pages 1–35, 2012. doi: 10.1007/s11042-012-1177-y.
- D. Salah und A. Zeid. Adaptive Patterns for Intelligent Tutoring Systems. In *Proceedings of the 14th European Conference on Pattern Languages of Programs*, 2009. URL [http://ceur-ws.org/Vol-566/d2b\\_Intelligent\\_Tutoring\\_Systems.pdf](http://ceur-ws.org/Vol-566/d2b_Intelligent_Tutoring_Systems.pdf) (abgerufen am 1.06.2016).
- P. Santos, D. Hernández-Leo, T. Navarrete und J. Blat. Computer Assisted Assessment Patterns for Different Target Users: The Case of Multimedia Questionnaires. In C. Kohls und J. Wedekind, editors, *Investigations of E-Learning Patterns: Context Factors, Problems and Solutions.*, pages 220–236. IGI Global, Hershey, Pennsylvania, 2011.
- N. Sawhney, D. Balcom und I. Smith. HyperCafe: Narrative and Aesthetic Properties of Hypervideo. In *Proceedings of the the seventh ACM conference on Hypertext HYPERTEXT 96*, volume 22, pages 1–10, 1996. doi: 10.1145/234828.234829.
- N. Schadewitz. Design Patterns for Cross-cultural Computer-supported Collaboration. *International Journal of Design*, 3(3):–, 2009. doi: 10.1.1.102.715.
- R. Schaefer. On the Limits of Visual Programming Languages. *SIGSOFT Softw. Eng. Notes*, 36(2):7–8, 2011. doi: 10.1145/1943371.1943373.
- S. Schaffert. *Strategic Integration of Open Educational Resources in Higher Education*, pages 119–131. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-03582-1. doi: 10.1007/978-3-642-03582-1\_11.
- R. C. Schank und R. P. Abelson. *Scripts, plans, goals and understanding / an inquiry into human knowledge structures*. Erlbaum, Hillsdale, NJ, 1977.
- S. Schön und M. Ebner. *Gute Lernvideos ... so gelingen Web-Videos zum Lernen!* Books on Demand, Norderstedt, 2013.
- U. Schonfeld und N. Shivakumar. Sitemaps: Above and Beyond the Crawl of Duty. In *18th International World Wide Web Conference*, page 991, apr 2009. doi: 10.1145/1526709.1526842.
- R. Schulmeister. Interaktivität in Multimedia-Anwendungen, 2005. URL <http://www.e-teaching.org/didaktik/gestaltung/interaktiv/InteraktivitaetSchulmeister.pdf> (abgerufen am 1.06.2016).
- R. Schulmeister. The Position of xMOOCs in Educational Systems. *eLeed*, 10(1), 2014. URL <http://nbn-resolving.de/urn:nbn:de:0009-5-40743> (abgerufen am 1.06.2016).
- P. Schultes, F. Lehner und H. Kosch. Videomarkup – Vom Betrachter zum Produzent. In *Workshop Audiovisuelle Medien WAM 2010*, pages 127–141, Chemnitz, 2010.
- T. Schümmer. *A Pattern Approach for End-User Centered Groupware Development*. Dissertation, FernUniversität Hagen, 2005.

- T. Schümmer. Best Practice for the pattern scout. In P. Goodyear und S. Retalis, editors, *Technology-Enhanced Learning – Design Patterns and Pattern Languages*, pages 255–275. Sense Publishers, 2010.
- T. Schümmer und S. Lukosch. *Patterns for computer-mediated interaction*. Wiley, Hoboken, NJ, 2007.
- S. Schwan. Gestaltungsanforderungen für Video in Multimedia-Anwendungen., 2005. URL <http://www.eteaching.org/didaktik/gestaltung/visualisierung/video/schwan.pdf> (abgerufen am 1.06.2016).
- S. Schwan und R. Riempp. The cognitive benefits of interactive videos: learning to tie nautical knots. *Learning and Instruction*, 14:293–305, 2004.
- D. L. Schwartz und K. Hartman. It is not television anymore: Designing digital video for learning and assessment. In R. Goldman, R. Pea, B. Barron und S. J. Derry, editors, *Video Research in the Learning Sciences*. Erlbaum, Mahwah, 2007.
- S. See und J. M. Conry. Flip My Class! A faculty development demonstration of a flipped-classroom. *Currents in Pharmacy Teaching and Learning*, 6(4):585–588, 2014. ISSN 18771297. doi: 10.1016/j.cptl.2014.03.003.
- N. Seel und D. Ifenthaler. *Online Lehren und Lernen*. Ernst Reinhardt Verlag, München, 2009.
- A. Seffah. The Evolution of Design Patterns in HCI: From Pattern Languages to Pattern-Oriented Design. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems*, pages 4–9, 2010. doi: 10.1145/1824749.1824751.
- A. Seffah und M. Taleb. Tracing the evolution of HCI patterns as an interaction design tool. *Innovations in Systems and Software Engineering*, 8(2):93–109, 2012. doi: 10.1007/s11334-011-0178-8.
- N. Seidel. *Web-basierte Hypervideo-Produktion*. Diplomarbeit, Universität Ulm, 2008.
- N. Seidel. Enable Wikis for Seamless Hypervideo Integration. In *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics*, pages 251–252, Rostock, Germany, 2011a. ACM. doi: 10.1145/2074712.2074765.
- N. Seidel. Hypervideo-Wissensmanagement am Beispiel einer internationalen Vorlesungsreihe. In K. Herring und J. Kawalek, editors, *WEL'11 - Workshop on eLearning*, pages 1–10, Leipzig, 2011b.
- N. Seidel. E-Learning-Modul on Integrated Water Resources Management: Konzepte und Werkzeuge für die Realisierung einer hypervideo-basierten Lernumgebung. In J. Desel, J. M. Haake und C. Spannagel, editors, *DeLFI 2012 – Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik*, pages 291–302, Bonn, 2012a. Gesellschaft für Informatik.
- N. Seidel. Collaborative hypervideo editing using MediaWiki. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, pages 22:1–22:4, Linz, 2012b. ACM. doi: 10.1145/2462932.2462962.
- N. Seidel. Interaction Design Patterns videographischer Lernmedien. In K. Herring und J. Kawalek, editors, *WEL'12 - Workshop on eLearning*, pages 37–48, Görlitz, 2012c.
- N. Seidel. Peer Assessment und Peer Annotation mit Hilfe eines videobasierten CSCL-Scripts. In A. Breiter und C. Rensing, editors, *DeLFI 2013 – Die 11. e-Learning Fachtagung Informatik der Gesellschaft für Informatik*, pages 83–94, Bonn, 2013. Gesellschaft für Informatik.
- N. Seidel. Analyse von Nutzeraktivitäten in linearen und nicht-linearen Lernvideos. *Zeitschrift für Hochschulentwicklung - Videos in der (Hochschul-)Lehre*, 9(3):164–186, 2014a. URL <http://www.zfhe.at/index.php/zfhe/article/view/687> (abgerufen am 1.06.2016).
- N. Seidel. Interaction design patterns for interactive video players in video-based learning environments. In *Proceedings of the 19th European Conference on Pattern Languages of Programs*, pages 19:1–19:14, New York, NY, USA, 2014b. ACM. doi: 10.1145/2721956.2721980.
- N. Seidel. Ein Untersuchungsdesign zum Vergleich von offener und Script-basierter Kollaboration beim Lernen mit Videos. In *DeLFI 2014 – Die 12. e-Learning Fachtagung Informatik der Gesellschaft für Informatik*, pages 297–300, Bonn, 2014c. Gesellschaft für Informatik.

- N. Seidel. Vergleich von offener und Script-basierter Kollaboration in einer Videolernumgebung. In T. Köhler und N. Kahnwald, editors, *GeNeMe'14 - Gemeinschaften in neuen Medien*, pages 93–104, Dresden, 2014d. TUDpress.
- N. Seidel. Interaction design patterns for design and development of video learning environments. In *Proceedings of the 19th European Conference on Pattern Languages of Programs*, pages 20:1–20:12, New York, NY, USA, 2014e. ACM. doi: 10.1145/2721956.2721981.
- N. Seidel. Das KZ-Außenlager Görlitz. In M. Bauer und S. Hoche, editors, *Die Juden von Görlitz - Beiträge zur jüdischen Geschichte von Görlitz*, pages 153–158. Verlag Gunter Oettel, Görlitz, 2014f.
- N. Seidel. Making web video accessible – interaction design patterns for assistive video learning environments. In *Proceedings of the 20th European Conference on Pattern Languages of Programs*, pages 17:1–17:16, New York, NY, USA, 2015a. ACM. doi: 10.1145/2855321.2855339.
- N. Seidel. Interaction design patterns for spatio-temporal annotations in video learning environments. In *Proceedings of the 20th European Conference on Pattern Languages of Programs*, pages 16:1–16:21, New York, NY, USA, 2015b. ACM. doi: 10.1145/2855321.2855338.
- N. Seidel. Video Assessment Patterns: Designing Video Player for Integrated Assessment. In *Proceedings of the 21st European Conference on Pattern Languages of Programs*, pages 32:1—32:20, New York, NY, USA, 2016. ACM. doi: 10.1145/3011784.3011820.
- N. Seidel und C. Jödicke. Aufgabentypen und Einsatzszenarien für ein E-Assessment in Lernvideos. In K. Herring und J. Kawalek, editors, *WEL'14 - Workshop on eLearning*, pages 83–92, Görlitz, 2014.
- F. Shipman, A. Girgensohn und L. Wilcox. Combining Spatial and Navigational Structure in the Hyper-Hitchcock Hypervideo Editor. In *HYPertext '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 124–125, New York, NY, USA, 2003a. ACM Press. doi: 10.1145/900051.900078.
- F. Shipman, A. Girgensohn und L. Wilcox. Hyper-Hitchcock: Towards the Easy Authoring of Interactive Video. In *Human-Computer Interaction INTERACT '03*, pages 33–40. IOS Press, 2003b.
- H. Shirouzu und N. Miyake. Scaffolds for Lecture Comprehension: Video-Jigsaw and Stop-and-Think Using a Video System. In *Proceedings of the 2007 Conference on Supporting Learning Flow Through Integrative Technologies*, pages 175–182, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.
- B. Shneiderman. *Designing the User Interface*. Addison-Wesley, Reading, Mass., USA, 1998.
- B. Shneiderman und C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Addison Wesley, Boston, MA, USA, 4. Auflage, 2004.
- H. A. Simon. The structure of ill structured problems. *Artificial Intelligence*, 4(3-4):181–201, 1973. doi: 10.1016/0004-3702(73)90011-8.
- C. Snelson. YouTube across the Disciplines: A Review of the Literature. *Journal of Online Learning and Teaching*, 7(1), 2011. URL [http://jolt.merlot.org/vol7no1/snelson\\_0311.htm](http://jolt.merlot.org/vol7no1/snelson_0311.htm) (abgerufen am 1.06.2016).
- L. F. G. Soares. Nested Context Language 3.0 Part 11 – Declarative Hypermedia Objects in NCL: Nesting Objects with NCL code in NCL Documents, 2009.
- R. J. Spiro, B. P. Collins und A. R. Ramchandran. Reflections on a post-Gutenberg epistemology for video use in ill-structured domains: Fostering complex learning and cognitive flexibility. In R. Goldman, R. D. Pea, B. Barron und S. Derry, editors, *Video research in the learning sciences*, pages 93–100. Lawrence Erlbaum Associates, Mahwah, NJ, 2007.
- H. Srl. Hyperfilm, 2007. URL <http://www.hyperfilm.eu/> (abgerufen am 1.06.2016).
- E. Stahl, C. Zahn, S. Schwan und M. Finke. Knowledge acquisition by hypervideo design: Different roles of writing during courses of "new"media production. In L. V. Waes, M. Leijten, C. Neuwirth und G. Rijlaarsdam, editors, *Studies in Writing: Vol. 17. Writing and digital Media*, volume 17, pages 77–88. Elsevier, Oxford, 2005.



- E. Stahl, M. Finke und C. Zahn. Knowledge acquisition by hypervideo design: an instructional program for university courses. *Journal of Educational Multimedia and Hypermedia*, 15(3):285–302, 2006. URL <http://www.editlib.org/p/6282> (abgerufen am 1.06.2016).
- P. D. Stotts und R. Furuta. Temporal hyperprogramming. *Journal of Visual Languages & Computing*, 1(3):237–253, 1990. doi: 10.1016/S1045-926X(05)80008-4.
- H. Suen. Peer Assessment for Massive Open Online Courses (MOOCs). *The International Review of Research in Open and distributed Learning*, 15(3), 2014. URL <http://www.irrodl.org/index.php/irrodl/article/view/1680/2904> (abgerufen am 1.06.2016).
- D. D. Suthers. *Scripting Computer-Supported Collaborative Learning: Cognitive, Computational and Educational Perspectives*, chapter Discussion, pages 177–187. Springer US, Boston, MA, 2007. doi: 10.1007/978-0-387-36949-5\_11.
- H.-E. Tenorth und R. Tippelt. *Beltz Lexikon Pädagogik - Studienausgabe*. Beltz, Langensalza, 1. Auflage, 2012.
- L. Tetzlaff und D. R. Schwartz. The Use of Guidelines in Interface Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 329–333, New York, NY, USA, 1991. ACM. doi: 10.1145/108844.108936.
- R. Thompson und C. J. Bowen. *Grammar of the Shot*. Focal Press, 2. Auflage, 2009.
- Thomson Reuters. Top 20 Countries in ALL FIELDS, 2001-August 31, 2011, 2011. URL <http://sciencewatch.com/articles/top-20-countries-all-fields-2001-august-31-2011> (abgerufen am 1.06.2016).
- J. Tidwell. Interaction Patterns. In *Proceedings of the 5th Conference on Pattern Languages of Programs*, 1998. URL [http://hillside.net/plop/plop98/final\\_submissions/P29.pdf](http://hillside.net/plop/plop98/final_submissions/P29.pdf) (22.05.2016) (abgerufen am 1.06.2016).
- J. Tidwell. *Designing Interfaces – Patterns for Effective Interaction Design*. O'Reilly Media, Sebastopol, 1. Auflage, 2005.
- J. Tidwell. *Designing Interfaces – Patterns for Effective Interaction Design*. O'Reilly Media, Sebastopol, 2. Auflage, 2011.
- C. A. B. Tiellet, A. Grahl Pereira, E. B. Reategui, J. V. Lima und T. Chambel. Design and Evaluation of a Hypervideo Environment to Support Veterinary Surgery Learning. In *HT'10*, pages 213–222, 2010.
- J. Tolva. MediaLoom: An Interactive Authoring Tool for Hypervideo, 1998. URL <http://www.ascentstage.com/medialoom/> (abgerufen am 1.06.2016).
- J. Tolva. MediaLoom: An Interactive Authoring Tool for Hypervideo. Bericht, Ascent Stage, 2007. URL <http://www.ascentstage.com/medialoom/> (abgerufen am 1.06.2016).
- S. Trahasch. *Skriptgesteuerte Wissenskommunikation und personalisierte Vorlesungsaufzeichnungen*. Logos Verlag, Berlin, 2006.
- D. Travis. 247 web usability guidelines, 2014. URL <http://www.userfocus.co.uk/resources/guidelines.html> (abgerufen am 1.06.2016).
- D. K. van Duyne, J. A. Landay und J. I. Hong. *The Design of Sites: Patterns for Creating Winning Web Sites*. Prentice Hall, 2. Auflage, 2006.
- M. van Welie. Patterns in Interaction Design, 2008. URL <http://www.welie.com/patterns/> (abgerufen am 1.06.2016).
- M. van Welie und B. Klaasse. Evaluation Museum Websites using Design Patterns. Bericht, Internal Report IR-IMSE-001, Faculty of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, 2004. URL <http://www.welie.com/papers/IR-IMSE-001-museum-sites.pdf> (abgerufen am 1.06.2016).
- M. van Welie und G. C. v. D. Veer. Pattern Languages in Interaction Design: Structure and Organization. In M. Rauterberg, editor, *Proceedings of Interact '03, 1-5 September*, pages 527–534, Amsterdam, 2003. IOS Press.

- J. Vantroys und Y. Peter. *Groupware: Design, Implementation, and Use: 9th International Workshop, CRIWG 2003, Autrans, France, September 28 - October 2, 2003. Proceedings*, chapter COW, a Fle, pages 168–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi: 10.1007/978-3-540-39850-9\_15.
- G. Vardi. Navigation scheme for interactive movies with linear narrative. In *HYPERTEXT '99: Proceedings of the tenth ACM Conference on Hypertext and hypermedia : returning to our diverse roots*, pages 131–132, New York, NY, USA, 1999. ACM. doi: 10.1145/294469.294500.
- C. Vaucelle, G. Davenport und T. Jehan. Textable movie: improvising with a personal movie database. In A. P. Rockwood, editor, *ACM SIGGRAPH 2003 Sketches & Applications*. ACM, 2003. URL <http://dblp.uni-trier.de/db/conf/siggraph/siggraph2003.html#VaucelleDJ03> (abgerufen am 1.06.2016).
- V. Venkatesh, M. G. Morris, G. B. Davis und F. D. Davis. User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3):425–478, 2003.
- E. Villasclaras-Fernández, D. Hernández-Leo, J. I. Asensio-Pérez und Y. Dimitriadis. Web Collage: An implementation of support for assessment design in CSCL macro-scripts. *Computers & Education*, 67:79–97, 2013. doi: 10.1016/j.compedu.2013.03.002.
- J. Vlissides. Seven Habits of Successful Pattern Writers. *C++ Report*, 8(10), 1996. URL <http://st-www.cs.illinois.edu/users/patterns/papers/7habits.html> (abgerufen am 1.06.2016).
- F. Vohle. Forschungsnotiz Nr. 3: Videoannotation. Ein Beitrag zur didaktischen Musterforschung? Bericht, Professur für Lehren und Lernen mit Medien, München, 2010.
- F. Vohle. Relevanz und Referenz: Zur didaktischen Bedeutung situationsgenauer Videokommentare im Hochschulkontext. Doppelfestschrift für Peter Baumgartner und Rolf Schulmeister. In G. Reinmann, M. Ebner und S. Schön, editors, *Hochschuldidaktik im Zeichen von Heterogenität und Vielfalt*, pages 165–182. Books on Demand GmbH, Norderstedt, 2013.
- F. Vohle und G. Reinmann. Social Video Learning and Social Change in German Sports Trainer Education. *International Journal of Excellence in Education*, 6(2), 2014. doi: 10.12816/0010834.
- I. Vosloo und D. G. Kourie. Server-centric Web Frameworks: An Overview. *ACM Computing Surveys*, 40(2):4:1–4:33, 2008. doi: 10.1145/1348246.1348247.
- W3C. Cascading Style Sheets (CSS) Snap Shoot 2010 – W3C Working Group Note 12 May 2011, 2011a. URL <http://www.w3.org/TR/CSS/> (abgerufen am 1.06.2016).
- W3C. Standard ECMA-262 ECMAScript Language Specification, 2011b. URL <http://www.ecma-international.org/ecma-262/5.1/> (abgerufen am 1.06.2016).
- W3C. Accessible Rich Internet Applications (WAI-ARIA) 1.0 W3C Candidate Recommendation 18 January 2011, 2011c. URL <http://www.w3.org/TR/2011/CR-wai-aria-20110118/> (abgerufen am 1.06.2016).
- C. Wallbaum. Wenn Musik nur in erfüllter Praxis erscheint. In C. Wallbaum, editor, *Perspektiven der Musikdidaktik. Drei Schulstunden im Licht der Theorien. Schriften / Hochschule für Musik und Theater "Felix Mendelssohn Bartholdy", Leipzig, 3*, pages 83–122. Olms, Hildesheim, 2010.
- K. Wannemacher. Digitale Lernszenarien im Hochschulbereich. Bericht, Geschäftsstelle Hochschulforum Digitalisierung, 2016. URL [http://www.che.de/downloads/HFD\\_AP\\_Nr\\_15\\_Digitale\\_Lernszenarien.pdf](http://www.che.de/downloads/HFD_AP_Nr_15_Digitale_Lernszenarien.pdf) (abgerufen am 1.06.2016).
- B. Weiberg. Beyond Interactive Cinema. *Interact*, pages 1–8, 1994. URL [www.keyframe.org/txt/interact/](http://www.keyframe.org/txt/interact/) (abgerufen am 1.06.2016).
- B. Weidenmann. Abbilder in Multimediaanwendungen. In L. J. Issing und P. Klimsa, editors, *Information und Lernen mit Multimedia und Internet*, pages 83–96. Beltz, Weinheim, 3. Auflage, 2002.

- A. Weinberger und F. Fischer. Computergestützte Kooperationsskripts. In J. Haake, G. Schwabe und M. Wessner, editors, *CSCL-Kompendium 2.0: Lehr- und Handbuch zum computerunterstützten, kooperativen Lernen*, chapter 3.5, pages 234–239. Oldenbourg Wissenschaftsverlag Verlag, München, 2. Auflage, 2012.
- A. Weinberger, B. Ertl, F. Fischer und H. Mandl. Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science*, 33(1):1–30, 2005. doi: 10.1007/s11251-004-2322-4.
- T. Wellhausen und A. Fiesser. How to Write a Pattern?: A Rough Guide for First-time Pattern Authors. In *Proceedings of the 16th European Conference on Pattern Languages of Programs*, EuroPLoP '11, pages 5:1–5:9, New York, NY, USA, 2012. ACM. doi: 10.1145/2396716.2396721.
- M. Wertheimer. *Produktives Denken*. Verlag Waldemar Kramer, Frankfurt/Main, 1964.
- M. Wessner. Werkzeuge für Scripted Collaboration. In J. Haake, G. Schwabe und M. Wessner, editors, *CSCL-Kompendium 2.0: Lehr- und Handbuch zum computerunterstützten, kooperativen Lernen*, volume 2., völlig, chapter 2.7, pages 159–162. Oldenbourg Wissenschaftsverlag, München, 2. Auflage, jan 2012. doi: doi:10.1524/9783486716825.159.
- A. Westbrook. Escaping the box: imagining a more visual internet. *Inside the story*, 1(2):41–48, 2013. URL <http://www.insidesthestory.org/issue2/> (abgerufen am 1.06.2016).
- C. Wetzels, P. Radtke und H. Stern. *Instructional effectiveness of video media*. Lawrence Erlbaum Associates, New Jersey Hove, UK, 1994.
- R. Wilkinson. Video-based learning activities. *TESL Canada Journal*, 1(2), 1984.
- Yahoo! ypatterns., 2012. URL <http://developer.yahoo.com/ypatterns/> (abgerufen am 1.06.2016).
- C. Zahn. *Dynamische Informationsvisualisierung in Videos und Animationen*. -, Leibnizinstitut für Wissenmedien, Tübingen, 2004.
- C. Zahn, F. Hesse, M. Finke, K. Media, R. Pea, M. Mills und J. Rosen. Advanced Digital Video Technologies to Support Collaborative Learning in School Education and Beyond. In *Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, pages 737–742. International Society of the Learning Sciences, 2005.
- C. Zahn, K. Krauskopf und F. W. Hesse. Video-Tools im Schulunterricht: Psychologisch-pädagogische Forschung zur Nutzung audiovisueller Medien. In *Workshop Audiovisuelle Medien WAM 2009*, pages 59–65, Chemnitz, 2009.
- C. Zahn, R. Pea, F. W. Hesse und J. Rosen. Comparing simple and advanced video tools as supports for complex collaborative design processes. *The Journal of the Learning Sciences*, 19(3):403–440, 2010. doi: 10.1080/10508401003708399.
- C. Zahn, K. Krauskopf, F. Hesse und R. Pea. How to improve collaborative learning with video tools in the classroom? Social vs. cognitive guidance for student teams. *International Journal of Computer-Supported Collaborative Learning*, 7(2):259–284, 2012. doi: 10.1007/s11412-012-9145-0.
- D. Zhang, L. Zhou, R. O. Briggs und J. F. N. Jr. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management*, 43(1):15–27, 2006. doi: 10.1016/j.im.2005.01.004.
- T. T. Zhou und J. S. Jin. Principles of Video Annotation Markup Language (VAML). *Reproduction*, pages 123–127, 2006.
- P. Zimbardo und R. J. Gerrig. *Psychologie*. Pearson Studies, München, 18. Auflage, 2009.



# Abbildungsverzeichnis

2.1	Gegenüberstellung von gängigen Repräsentationen und Aufnahmeformaten von Lernvideos . . . . .	31
2.2	Ein Beispiel eines Lernvideos ohne und mit einer Hervorhebung . . . . .	39
2.3	Ein Beispiel eines Lernvideos mit und ohne der räumlichen Nähe von Bild und Text	41
2.4	Beispiel eines Lernvideos, bei dem Formeln per Stifteingabe auf einem <i>tablet computer</i> zusammen mit dem Tonsignal aufgezeichnet wurden . . . . .	42
2.5	Schematische Übersicht der vier grundlegenden videobezogenen Lernformen in Anbetracht der Interaktion mit anderen Lernenden und dem Videosystem . . . .	48
3.1	Merkmale verschiedener Formen von nicht-linearen Videos im Überblick . . . . .	63
3.2	Ein Beispiel für die zeitlichen Beziehungen zwischen einem Video und vier synchronisierten Folien . . . . .	67
3.3	Unterscheidung von Hyperlinks anhand ihrer Verweisziele . . . . .	72
3.4	Zwei Beispiele für zeitleistenbasierte Autorensysteme . . . . .	84
4.1	Mehrfachzuordnungen der Gruppen von Informationsträgern . . . . .	114
4.2	Exemplarische Eingabemaske eines kodierten Informationsträgers . . . . .	122
4.3	Verteilung der Anzahl der pro Informationsträger identifizierten Kategorien . . . .	125
4.4	Verteilung der Anzahl an Informationsträgern je Kategorie . . . . .	125
4.5	Häufigkeitsverteilung der Kategorien in den Analysedimensionen je Gruppe von Informationsträgern . . . . .	127
4.6	Gesamtübersicht der Vorgehensweise zur Elaboration von Interaction Design Patterns . . . . .	139
4.7	<i>Pattern Map</i> . . . . .	142
5.1	Videobasierte CSCL-Scripts als Schnittmenge videobasierter Scripts und CSCL-Scripts . . . . .	176
5.2	ER-Diagramm der Script-Komponenten und ihrer Beziehungen untereinander . .	199
5.3	S-BPM Graphennotation eines vereinfachten Peer Assessment Scripts . . . . .	204

5.4	Grafische Darstellung des Hypervideo-Dokuments . . . . .	207
5.5	Grafische Darstellung des Datenmodells für ein Script-Template . . . . .	208
5.6	Entscheidungen bei der Modellierung von Scripts der Komplexitätsstufen 0 – 3. . . . .	210
5.7	Visualisierung des zeitlichen Verlaufs der Script-Phasen . . . . .	211
5.8	Sankey-Diagramm der über drei Phasen wechselnden Gruppenkonstellationen . . . . .	211
6.1	Architektur von <i>VI-TWO</i> . . . . .	223
6.2	Klassifikation von Videoannotationen hinsichtlich des zeitlichen und räumlichen Layouts . . . . .	225
6.3	Fließdiagramm der Verarbeitung deklarativer Auszeichnungen für die Repräsentation innerhalb von <i>VI-TWO</i> . . . . .	226
6.4	Schematische Darstellung der im Widget <i>vi2.Hyperlinks</i> implementierten Hyperlink-Varianten . . . . .	228
6.5	Suchfunktion in <i>IWRM education</i> . . . . .	231
6.6	Eingabedialog in <i>Vi-Wiki</i> . . . . .	234
6.7	Integration des <i>Mozilla Popcorn Makers</i> . . . . .	235
6.8	Client-Server-Architektur von <i>VI-LAB</i> . . . . .	238
6.9	Übersicht der Videoinstanzen im Administrationsbereich von <i>VI-LAB</i> . . . . .	241
6.10	GUI für die Erstellung und Bearbeitung eines Script-Templates . . . . .	243
6.11	GUI für die Bearbeitung von Script-Instanzen auf dem Komplexitätslevel 2 im Administrationsbereich von <i>VI-LAB</i> . . . . .	243
6.12	<i>Absolute Retention Rate</i> je Video . . . . .	245
6.13	<i>Rewatching Graph</i> . . . . .	245
6.14	<i>CORDTRA</i> -Diagramm . . . . .	246
A.1	Übersicht über die Browser-Versionen, die das HTML5 Video-Element unterstützen. . . . .	267
A.2	<i>Shaded Similarity Matrix</i> von 40 Videolernumgebungen . . . . .	294
A.3	Dendogramm der hierarchischen Clusterisierung der Informationsträger hinsichtlich der Anzahl gemeinsamer Kategorien. . . . .	296
A.4	Ergebnisse der Pattern-Evaluation . . . . .	307

# Tabellenverzeichnis

1.1	Übersicht der Methologie je Kapitel . . . . .	28
2.1	Aus den 12 Prinzipien des kognitiven Lernens mit Multimedia abgeleitete Anforderungen an Videolernumgebungen. . . . .	44
2.2	Konzepte des selbstgesteuerten Lernens . . . . .	51
2.3	Exemplarische Ausprägungen des kollaborativen Lernens . . . . .	52
3.1	Übersicht über LMS und ihre Funktionalität in Bezug auf Lernvideos . . . . .	61
3.2	Die sechs Stufen der Interaktivität in Relation zu den Hypervideo-Eigenschaften .	64
3.3	Die 13 möglichen zeitlichen Relationen zwischen zwei zeitlichen Intervallen . . .	68
4.1	Vergleichende Darstellungen verschiedener Gestaltungsrichtlinien . . . . .	107
4.2	Übersicht grundlegender Methoden des <i>Pattern Minings</i> . . . . .	112
4.3	Kriterien zur Auswahl der Informationsträger . . . . .	115
4.4	Gruppen von Informationsträgern . . . . .	116
4.5	Kategorien je Analysedimension . . . . .	118
4.6	Vergleich der Analysedimensionen mit Angabe der Anzahl an zugehörigen Kategorien. . . . .	126
4.7	45 Kategorien von UI-Komponenten von Videolernumgebungen je Dimension . .	127
4.8	Pattern Languages, aus denen Muster adaptiert werden konnten . . . . .	129
4.9	Verteilung der Patterns auf die neun Layer des <i>Universal Model of User Interface</i>	137
4.10	Gliederung der Pattern Language in Mikro- und Makrointeraktivität sowie hinsichtlichlich funktionaler Aspekte. . . . .	141
4.11	45 Interaction Design Patterns für Videolernumgebungen . . . . .	142
4.12	Funktionen der Wissensbasis im Überblick . . . . .	146
4.13	Übersicht der ausgewählten Design Patterns . . . . .	151
5.1	Übersicht von Beispielen videobasierter CSCL-Scripts . . . . .	178
5.2	Exemplarischer Ablauf eines PAPA-Scripts. . . . .	180



5.3	Ablauf eines Scripts für Videoanalyse und -diskussion in der minimalen Konfiguration . . . . .	181
5.4	Übersicht von Aktivitäten unter Angabe der jeweiligen Ein- und Ausgangsgrößen unter Berücksichtigung von Interaction Design Patterns und <i>thinkLets</i> . . . . .	189
5.5	Komplexitätslevel videobasierter CSCL-Scripts . . . . .	200
6.1	Übersicht der Anforderungen (R) an ein Framework für interaktive Videos . . . . .	218
6.2	Bewertung verwandte Frameworks hinsichtlich der an sie gestellten Anforderungen . . . . .	221
6.3	Vergleich von Ansätzen für die Erstellung, Bearbeitung und Verwaltung von Videoannotationen . . . . .	234
6.4	Übersicht der Anforderungen (RS) an <i>VI-LAB</i> . . . . .	236
6.5	In den Anwendungen <i>Vi-Wiki</i> , <i>IWRM education</i> sowie den unterschiedlichen Entwicklungsstufen von <i>VI-LAB</i> realisierte Interaction Design Patterns . . . . .	251
A.1	Usability Guidelines: Design . . . . .	262
A.2	Usability Guidelines: Navigation . . . . .	263
A.3	Usability Guidelines: Suche . . . . .	263
A.4	Usability Guidelines: Technologie . . . . .	264
A.5	Usability Guidelines: Benutzerfeedback und Fehlermeldungen . . . . .	264
A.6	Usability Guidelines: Zeitabhängige Objekte und Bedienelemente . . . . .	265
A.7	Usability Guidelines: Barrierefreiheit . . . . .	266
A.8	Unterstützung gängiger Video-Codecs durch verbreitete Web-Browser . . . . .	267
A.9	Spezifische Suchbegriffe je Rechercheplattform . . . . .	268
A.10	Ergebnisse der Inhaltsanalyse durch den zweiten Schätzer . . . . .	288
A.11	Vergleich der Analysedimensionen bezogen auf die zugehörigen Kategorien . . . . .	291
A.12	Vergleich der Analysedimensionen bezogen auf die zugehörigen Kategorien (Fortsetzung) . . . . .	292
A.13	Vergleich der Analysedimensionen bezogen auf die zugehörigen Kategorien (Fortsetzung) . . . . .	293
A.14	Art der Beziehung zwischen den identifizierten Mustern und den Mustern in anderen Musterkollektionen . . . . .	306
A.15	Den Patterns zugeordnete Layer des <i>Universal Model of User Interface</i> . . . . .	308
A.16	Binäres Entscheidungsmodell zur Suche und Auswahl von Design Patterns . . . . .	310
A.17	Evaluationsergebnisse und Bewertung der Videolernumgebung <i>eKomMed.</i> . . . .	312
A.18	Beispiele für Scripts und CSCL-Scripts . . . . .	335

A.19 Spezifikation benutzerdefinierter HTML data-Attribute in VI-TWO . . . . .	343
--	-----



# Verzeichnis der Auflistungen

3.1	Schemata und Beispiele der <i>Media Fragment URI</i> . . . . .	79
3.2	YouTube URI-Schema für Medienfragmente unter Angabe der eindeutigen Video-Id sowie den Minuten (<m>) und Sekunden (<s>) der Abspielposition . . . . .	80
6.1	Beispielcode für das Widget vi2.RelatedVideos in VI-TWO . . . . .	232
6.2	In VI-TWO eingeführte HTML-Elemente ( <i>Costume Elements</i> ) . . . . .	233
6.3	Erweiterung der <i>MediaWiki markup specification</i> . . . . .	233
6.4	Exemplarische HTTP-Routen in <i>VI-LAB</i> . . . . .	239
A.1	<i>Document Type Definition</i> der Pattern Language Markup Language . . . . .	318
A.2	JSON-Adaption der PLML als <i>Mongoose Schema</i> Definition . . . . .	319
A.3	Exemplarische Klasse zur Initialisierung von VI-TWO . . . . .	341
A.4	Exemplarisches Widget für VI-TWO . . . . .	342



# Verzeichnis der Algorithmen

1	Ein- und Ausblendung von Annotationen in Abhängigkeit der Wiedergabezeit je Zeitintervall der Prüfung . . . . .	78
2	Anzeigesteuerung zeitabhängiger Annotationen mit Hilfe einer indexbasierten Suche . . . . .	228





## Abkürzungsverzeichnis

AICC	Aviation Industry Computer-Based Training Committee – Organisation
AJAX	Asynchronous JavaScript and XML – Programmier Technik
API	Application Programming Interface – Programmier Technik
BMBF	Bundesministerium für Bildung und Forschung – Organisation
BMU	Bundesministerium für Umwelt und Verbraucher – Organisation
BSON	Binäre Erweiterung von → <i>JSON</i> – Datenformat
CBT	<i>Computer Based Training</i> – Lernformat
CMS	<i>Content Management System</i> – Software
CRUD	Abkürzung für die Operationen <i>Create, Read, Update, Delete</i>
CLFP	Collaborative Learning Flow Patterns
CSS	<i>Cascading Style Sheet</i> – Webstandard
CSCL	<i>Computer Supported Collaborative Learning</i> – Forschungsbereich
CSV	<i>Comma-separated values</i> – Dateiformat
ECMAScript	Scriptsprache der <i>European Computer Manufacturers Association</i> , auch bekannt als JavaScript.
EML	Educational Modeling Language – Datenmodell
ER-Model	<i>Entity Relation Model</i> – Datenmodell
GET	HTTP-Anfragemethode, → HTTP
GIFT	Datenformat für E-Assessment Aufgaben – Datenformat
GUI	Grafical User Interface
HCI	Human Computer Interaction – Forschungsbereich
IFrame	<i>inline frame</i> – HTML-Element
IMS	<i>Instructional Management Systems</i> – Organisation
IMS LD	<i>IMS Learning Design</i> – Datenmodell
IWRM	<i>Integrated Water Resources Management</i> – Forschungsbereich
JSON	<i>JavaScript Object Notation</i> – Datenformat
HATEOAS	<i>Hypermedia as the Engine of Application State</i> – Datenmodell
HTML	<i>Hyper Text Markup Language</i> – Webstandard
HTTP	<i>Hyper Text Transfer Protocol</i> – Internetprotokoll
LCMS	<i>Learning Content Management System</i> – Lernmanagementsystem
LD	<i>Learning Design</i> , → <i>IMS LD</i>
LMS	<i>Learning Management System</i> – Software
LOM	<i>Learning Object Model</i> – Datenmodell
MOOC	<i>Massive Open Online Course</i> – Lernformat
PLoP	<i>Pattern Languages of Programs</i> – Konferenzreihe
POST	HTTP-Anfragemethode, → HTTP
QTI	<i>IMS Question &amp; Test Interoperability Specification</i> – Datenmodell
REST	<i>Representational State Transfer</i> – Programmierparadigma
RSS	<i>Rich Site Summary</i> – Datenformat
S-BPM	<i>Subject-orientated Business Process Model</i> – Datenmodell
SCORM	<i>Sharable Content Object Reference Model</i> – Datenmodell.
SMIL	<i>Synchronized Multimedia Integration Language</i> – Datenformat

<b>SEO</b>	<i>Search Engine Optimization</i>
<b>SOA</b>	<i>Service Oriented Architectures – Software Architektur Pattern</i>
<b>UI</b>	<i>User Interface</i>
<b>UNESCO</b>	<i>United Nations Educational, Scientific and Cultural Organization – Organisation</i>
<b>URI</b>	<i>Uniform Resource Identifier</i>
<b>URL</b>	<i>Uniform Resource Location</i>
<b>UX</b>	<i>User Experience</i>
<b>W3C</b>	<i>World Wide Web Consortium – Organisation</i>
<b>WAI-ARIA</b>	<i>Web Accessibility Initiative – Accessible Rich Internet Applications – Web-standard</i>
<b>WBT</b>	<i>Web Bases Training – Lernformat</i>
<b>WWW</b>	<i>World Wide Web</i>
<b>XML</b>	<i>Extensible Markup Language – Datenformat</i>
<b>xMOOC</b>	Variante eines → <i>MOOC</i> . Das „x“ steht dabei für <i>extension</i> . – Lernformat